

# MACHINE LEARNING METHODS FOR MACHINE TEACHING

A Dissertation

Presented to the Faculty of the Graduate School  
of Cornell University

in Partial Fulfillment of the Requirements for the Degree of  
Doctor of Philosophy

by

Igor Labutov

August 2016

© 2016 Igor Labutov  
ALL RIGHTS RESERVED

## MACHINE LEARNING METHODS FOR MACHINE TEACHING

Igor Labutov, Ph.D.

Cornell University 2016

Over the course of the last five years, the Internet has demonstrated the potential of scaling learning beyond the walls of the traditional, physical classrooms. Massive Open Online Courses (MOOCs) — virtual classrooms that offer worlds' top university courses to anyone with an Internet connection — have captured the imagination of the public with their potential to bring world class education to the underserved regions and communities around the world.

The recent years of piloting the first MOOCs have also highlighted the challenges involved in scaling learning to the web, e.g., efficiently grading and providing feedback in classes with thousands of students. Niche communities of researchers and practitioners have formed around these core problems of *learning at scale*, which lies at the intersection of data mining, machine learning and education. Much progress has been made in developing data-driven techniques (e.g., peer-grading) that address the scalability barriers of the traditional means (e.g., for assessment and feedback).

The thrust of the recent progress in addressing the scalability barrier in education is summarized well with the following quotation:

*“At every order of magnitude expect every process to break. You would need to change the process each time at each level of scale.”* (Eric Schmidt)

This quote frames scaling as an inherent problem that requires developing solutions for dealing with it. To this end, much of the initial effort in *learning at*

*scale* has been directed precisely towards extending the limits of the traditional work-flows that arise as a consequence of scaling (e.g., assignment grading).

In this thesis, we instead take an alternative perspective on scaling — as an opportunity to develop learning tools whose existence is only afforded by classrooms that grow sufficiently large. This thesis presents a collection of models and algorithms that leverage the scaling phenomenon in order to develop tools to open new opportunities in making learning and teaching more effective. The overarching theme of this thesis is *democratization* — the idea that by blurring the distinction between learners and instructors, we create an opportunity to leverage the large number of learners to steer their own learning experience (e.g., by contributing to the learning content). The value of “letting everyone contribute” is amplified as classrooms grow bigger and more diverse, e.g., diverse in the learners’ backgrounds, learning styles and learning goals. Blurring the boundary between learners and instructors provides not only a solution to the scaling problem in tasks like grading, but also creates an opportunity to make the learning experience more attuned to the diversity of the learners.

Delegating learners with an active role in shaping their learning experience, e.g., via creating original learning content, creates a technical challenge of automatically “filtering” students’ contributions in order to distill valuable content that can be safely integrated into the learning experience. In this thesis, we investigate two core components that comprise a learning experience: (i) generation and sequencing of learning material and (ii) assessment of students’ learning. For both components, we develop and evaluate a number of machine learning models that leverage students’ contributions and interactions in the classroom in order to diversify and personalize students’ learning experience.

## BIOGRAPHICAL SKETCH

Igor Labutov was born in the city of Nizhny Novgorod, Russia (former Gorky, USSR), where he attended the math and physics lyceum #82 until the 5th grade. He and his family then immigrated to New York City, where he continued his education, attending the Stephen A. Halsey Junior High School #157, Forest Hills High School and finally earning his Bachelor's degree in Computer Engineering from the City College of New York. He then came to Cornell to pursue his PhD, where his research was advised at different stages by Professor Christoph Studer and Professor Hod Lipson. Beginning in August of 2016, he is starting a postdoctoral position in the Machine Learning department at Carnegie Mellon University.

To my Mom, Dad and Bishan.

## ACKNOWLEDGEMENTS

First, I would like to thank my advisor Professor Christoph Studer and my former advisor Professor Hod Lipson, who had provided invaluable support through the different stages of my six years at Cornell. I am thankful to both Christoph and Hod for sharing their boundless research energy, ideas, inspiration and most importantly for always giving their time to listen, understand and help. I would not get to this point without their support.

I am also extremely grateful to Professor Thorsten Joachims, who in addition to being an invaluable research collaborator and a mentor, provided me with the opportunity to TA his Machine Learning course — one of my most rewarding and formative experiences at Cornell. I am also grateful for Thorsten’s advise, which on a number of occasions helped me tremendously in important decisions.

I would like to thank Lucy Vanderwende and Sumit Basu — my mentors at Microsoft Research, with whom I had an incredibly fun and productive internship in the Summer of 2014. Most importantly, however, I am grateful for the time and energy they generously invested in me well outside the scope of the internship. I am especially thankful to Sumit and Lucy for giving me a better understanding of myself through advise that will likely serve me throughout my career.

I would like to thank my research collaborators: Sid Reddy, Frans Schalekamp, Kelvin Luu and Professor Siddhartha Banerjee. Aside from their ideas and insights that in many ways contributed to the work in this dissertation, they have also been a continuous source of inspiration and motivation in my research.

This acknowledgement would be far from complete without my closest Ithaca friends, who have made my experience at Cornell one that I will always cherish. I am thankful to Rina Tse, Apoorva, Maia Kelner and Andrey Gushchin. These people have provided unconditional support in sometimes challenging times,

and without question had left a fundamental and lasting influence on me.

I would also like to thank several incredible people who worked at Collee-town Bagels (CTB) during my time in Ithaca who became my friends. Despite the Ithacan winters, they consistently created a warming and welcoming atmosphere, where many of the ideas in this dissertation were born. I would like to especially thank Alana, Angelica, Cindy, Frank, Jackson, Regan and Thad.

Finally, I would like to thank the most important people in my life — my Mom, my Dad and Bishan, who have always supported me and sacrificed unconditionally. This dissertation is dedicated to them.

I am thankful to Bishan for her unconditional love, patience and support whose extent was beyond anything that I can put in words. Her encouragement, dedication and care turned the last six years into an incredible adventure. I cannot imagine doing it without her and I cannot wait for our many adventures ahead.

I would like to thank my Mom, Galina Labutova, and my Dad, Igor Y. Labutov, for their effort and sacrifices in bringing me to the United States, where I had the opportunity to pursue my dreams. I am eternally grateful to them for encouraging me to explore and pursue my many interests during my childhood, which had inevitably led me towards discovering my passion for research. Their relentless love and patience made me into the person I am today. I thank my Mom for making the right and difficult decisions regarding my education at the right time — without doubt they became pivotal in getting me to this stage. I thank my Dad for getting deeply involved and taking seriously every one of my curiosities and interests — no matter how ridiculous — from blowing up hydrogen in the kitchen, to building wings for my scooter in my attempt to fly.

## TABLE OF CONTENTS

Biographical Sketch . . . . .	iii
Dedication . . . . .	iv
Acknowledgements . . . . .	v
Table of Contents . . . . .	vii
List of Tables . . . . .	xii
List of Figures . . . . .	xiii
<b>1 Introduction</b>	<b>1</b>
1.1 Contributions of this dissertation . . . . .	3
1.1.1 Assessment . . . . .	3
1.1.2 Learning content curation . . . . .	7
1.2 Summary of contributions by research collaborators . . . . .	9
<b>2 Background</b>	<b>11</b>
2.1 Assessment . . . . .	11
2.1.1 Psychometrics . . . . .	12
2.1.2 Adaptive testing . . . . .	13
2.1.3 Learning at scale . . . . .	14
2.1.4 Crowdsourcing . . . . .	17
2.2 Learning content . . . . .	19
2.2.1 Early Intelligent tutoring systems (1920s-1990s) . . . . .	20
2.2.2 Modern Intelligent tutoring systems (1990s-2016) . . . . .	23
2.2.3 Hypermedia . . . . .	24
<b>3 Question generation from text</b>	<b>26</b>
3.1 Introduction . . . . .	26
3.2 Related Work . . . . .	29
3.3 An ontology of categories and sections . . . . .	31
3.4 Crowdsourcing methodology . . . . .	33
3.4.1 Question generation task . . . . .	34
3.4.2 Question relevance rating task . . . . .	35
3.5 Model . . . . .	36
3.5.1 Category/section inference . . . . .	37
3.5.2 Relevance classification . . . . .	38
3.6 Experiments and results . . . . .	40
3.6.1 Dataset I: for relevance classification . . . . .	40
3.6.2 Dataset II: for End-to-end evaluation . . . . .	41
3.6.3 Information Retrieval-based evaluation . . . . .	41
3.7 Examples and error analysis . . . . .	44
3.8 Conclusion . . . . .	46

<b>4</b>	<b>Optimal assessment with multiple choice questions</b>	<b>48</b>
4.1	Introduction . . . . .	48
4.2	Related Work . . . . .	49
4.2.1	Education . . . . .	49
4.2.2	Active Learning and Adaptive Testing . . . . .	50
4.3	Optimization with discrete choice models . . . . .	51
4.4	Model . . . . .	51
4.4.1	Relationship to the Rasch model . . . . .	54
4.4.2	Relationship to Bachrach et al. . . . .	55
4.5	Optimal Choice Sets . . . . .	57
4.5.1	Asymptotically optimal choices . . . . .	59
4.6	Algorithm for finding optimal choice sets . . . . .	62
4.7	Synthetic Experiments . . . . .	66
4.7.1	Parameter learning . . . . .	67
4.7.2	Optimal choice sets . . . . .	68
4.8	User Study: "US States Quiz" . . . . .	70
4.8.1	Data collection . . . . .	71
4.8.2	Evaluation . . . . .	72
4.9	Results . . . . .	73
4.9.1	Within-subject correlation . . . . .	73
4.9.2	Between-subject correlation . . . . .	74
4.10	Crowdsourcing tests from forums . . . . .	76
4.10.1	Modeling users and questions . . . . .	79
4.10.2	Generative Model and Inference . . . . .	80
4.10.3	Examples . . . . .	81
4.11	Discussion . . . . .	83
4.12	Future Work . . . . .	83
4.12.1	Multidimensional extension . . . . .	86
4.13	Appendix . . . . .	90
4.13.1	Derivation of the optimal choice set (scalar case) . . . . .	90
4.13.2	Derivation of the optimal choice set (multidimensional case) . . . . .	91
4.13.3	Optimal choice set when $\theta = 0$ . . . . .	92
<b>5</b>	<b>Joint Assessment and Grading</b>	<b>95</b>
5.1	Introduction . . . . .	95
5.1.1	JAG: An alternative to Peer Grading . . . . .	96
5.2	Related Work . . . . .	98
5.2.1	Crowdsourcing . . . . .	98
5.2.2	Peer-grading . . . . .	99
5.2.3	Clustering submissions . . . . .	100
5.3	Model . . . . .	101
5.3.1	Fully observed setting . . . . .	101
5.3.2	Partially observed setting . . . . .	102
5.4	Parameter Learning . . . . .	107

5.5	Experiments with Synthetic Data . . . . .	109
5.5.1	Simulation procedure . . . . .	110
5.6	Real-World Experiments . . . . .	111
5.6.1	Results . . . . .	113
5.7	Conclusion . . . . .	117
<b>6</b>	<b>Latent Choice Allocation: solution clustering from implicit feedback</b>	<b>120</b>
6.1	Introduction . . . . .	120
6.1.1	Multiple choice testing . . . . .	121
6.1.2	Peer-grading . . . . .	122
6.1.3	Proposed framework and contributions . . . . .	124
6.2	LCA: Latent Choice Allocation . . . . .	125
6.2.1	Model without latent answers . . . . .	126
6.2.2	Model with latent answers . . . . .	127
6.2.3	Dirichlet process (DP) prior for latent answers . . . . .	131
6.2.4	Parameter inference . . . . .	134
6.3	Synthetic Experiments . . . . .	134
6.3.1	Experiment 1: Extreme configurations . . . . .	134
6.3.2	Experiment 2: Effect of hyper-parameters . . . . .	136
6.4	Real-World Experiments . . . . .	138
6.4.1	Experiment setup . . . . .	138
6.4.2	Accuracy results . . . . .	141
6.4.3	Clustering results . . . . .	141
6.5	Related Work . . . . .	146
6.5.1	Student submission clustering . . . . .	146
6.5.2	Peer-grading and crowdsourcing . . . . .	147
6.5.3	Independence of irrelevant alternatives . . . . .	148
6.6	Conclusion and Future Work . . . . .	149
<b>7</b>	<b>Calibrated Self-grading</b>	<b>151</b>
7.1	Introduction . . . . .	151
7.2	Related Work . . . . .	153
7.3	Model . . . . .	155
7.3.1	Parameter estimation . . . . .	157
7.3.2	Calibrating the model . . . . .	159
7.3.3	Consequences of students' awareness of the mechanism . . . . .	161
7.4	Experiments . . . . .	164
7.4.1	Simulations . . . . .	164
7.4.2	User study . . . . .	169
7.4.3	Self-assessment and bias . . . . .	171
7.5	Conclusion and Future Work . . . . .	172

<b>8</b>	<b>Learning vocabulary from reading the web</b>	<b>174</b>
8.1	Introduction . . . . .	174
8.2	Related Work . . . . .	176
8.2.1	Incidental vocabulary acquisition . . . . .	177
8.2.2	Code-switching as a natural phenomenon . . . . .	178
8.2.3	Computational approaches to code-switching . . . . .	178
8.2.4	Computational approaches to sentence simplification . . . . .	179
8.2.5	Recent neurophysiological findings . . . . .	180
8.3	Model (part 1) . . . . .	181
8.4	User studies . . . . .	182
8.4.1	Corpus . . . . .	182
8.4.2	Study I: Modelling word acquisition rate . . . . .	184
8.4.3	Study II: Modelling effect of context . . . . .	187
8.5	Model (part 2) . . . . .	190
8.6	Experiments . . . . .	193
8.7	Conclusion and Future work . . . . .	196
<b>9</b>	<b>Learning content representation for personalized lesson sequencing</b>	<b>198</b>
9.1	Introduction . . . . .	198
9.2	Our Contributions . . . . .	199
9.3	Embedding Model . . . . .	200
9.3.1	Modeling Assessment Results . . . . .	202
9.3.2	Modeling Student Learning from Lessons . . . . .	203
9.4	Parameter Estimation . . . . .	205
9.5	Experiments on Synthetic Data . . . . .	206
9.6	Experiments on Online Course Data . . . . .	210
9.6.1	Assessment Result Prediction . . . . .	210
9.6.2	Lesson Sequence Discrimination . . . . .	214
9.7	Conclusion . . . . .	219
<b>10</b>	<b>Curating targeted learning paths through the web</b>	<b>221</b>
10.1	Introduction . . . . .	221
10.2	Related work . . . . .	224
10.3	Model . . . . .	226
10.3.1	Semi-supervised learning of term aspect . . . . .	229
10.4	Optimal learning paths . . . . .	233
10.5	Problem hardness . . . . .	236
10.6	ILP formulation . . . . .	240
10.7	Variation: Layered Set-Cover . . . . .	242
10.8	Experiments . . . . .	249
10.8.1	Prerequisites . . . . .	249
10.8.2	Scaling to the web . . . . .	253
10.8.3	Diversity of assumptions . . . . .	254
10.8.4	Fundamental prerequisites . . . . .	255

10.9 Conclusion . . . . .	263
<b>11 Conclusion and future work</b>	<b>265</b>
11.1 Summary of contributions . . . . .	266
11.1.1 Assessment . . . . .	266
11.1.2 Learning content curation . . . . .	268
11.2 Future directions . . . . .	269
11.2.1 Growth of implicit assessment . . . . .	269
11.2.2 Social organization of crowdsourcing efforts for learning content on the web . . . . .	271
<b>Bibliography</b>	<b>274</b>

## LIST OF TABLES

3.1	Most frequent section titles by category. . . . .	32
3.2	Examples of retrieved questions. TP, TN, FP, FN stand for true/false positive/negative with respect to the relevance classification. . . . .	45
8.1	A categorization shown to the worker during the tutorial to guide in evaluating the quality of their guess. . . . .	187
9.1	Test AUC, validation AUC, and standard error of validation AUC for variants of the LSE model and benchmark IRT models. . . . .	211

## LIST OF FIGURES

1.1	A space of all <i>assessment frameworks</i> (which broadly encompass both, the process for generating assessment content and administering it), can be represented in terms of the roles delegated to the instructors, peers and automation. For example, traditional tests generated and graded by instructors would at the lower left corner. The contribution of the first part of this thesis lies in methods for scalable assessment <i>generation</i> and <i>administration</i> . . .	4
3.1	Overview of our ontology-crowd-relevance approach. . . . .	27
3.2	Coverage properties of our category-section representation: (a) fraction of Wikipedia articles covered by the top $j$ most common Freebase types, grouped by our eight higher-level categories. (b) Average fraction of sections covered per document if only the top $k$ most frequent sections are used; each line represents one of our eight categories. . . . .	30
3.3	Prompt for the generation task for the category-section pair (Person, Legacy). . . . .	33
3.4	Average relevance and scope of worker-generated questions versus how the workers were prompted. . . . .	35
3.5	ROC curves for the task of question-to-article relevance prediction. $T_n$ means that $n$ positively labeled article segments were available for each question template during training. . . . .	39
3.6	Precision-recall results for the end-to-end experiment, grouped in bins of recall ranges. . . . .	44
4.1	Geometric intuition behind the choice of an optimal set of options in a multiple choice test: we imagine the subject's distance from the wall to be inversely related to their "ability to see." If asked which colored dot painted on the wall is the right-most dot, subjects closer to the wall would be more likely to answer the question correctly, and subjects farther away would be most likely to guess. The question of optimal choice set design can then be posed as: "where on the wall do we paint the dots, to most efficiently learn about your distance to the wall?" . . . . .	59

4.2	Performance (simulation) of the model in (a) predicting the correct answers in a set of questions and (b) ranking students by their ability, as a function of (i) number of choices shown in each question and (ii) variance of the choice parameter distribution (shown as standard deviation $\sigma_\beta$ ). “Easier” questions correspond to those with a “wider” spread between choice parameters (i.e., higher variance). We can conclude the following on the basis of these results: (i) more choices improves performance in both, predicting correct answers and ranking students, but with diminishing returns, and (ii) showing “easier” questions generally improves performance in correct answer prediction and ranking, however, the model is able to rank well even when the correct answers are more difficult to identify (see Section 4.7). Note that the random baseline for accuracy in (a) is 5% as there are 20 choices for each question in the simulation. . . . .	66
4.3	Rank correlation between the true and inferred student rankings as a function of the number of questions answered by each simulated student, separated by the choice sampling strategy. Optimizing choice sets according to the proposed objective ( <b>OPT-average</b> and <b>OPT-individual</b> ) results in better rank correlation with fewer questions compared to when the choice sets are sampled randomly. Optimizing choice sets according to the individual student abilities ( <b>OPT-individual</b> ) marginally improves performance over optimizing choice sets based on the average student ability ( <b>OPT-average</b> ). . . . .	69
4.4	Within-subject correlation between raw scores attained on the <b>subsetMCQ</b> and <b>fullMCQ</b> tests separated by choice set design strategy—choice sets optimized according to the proposed objective yield better within-subject score correlation than choice sets sampled randomly. . . . .	74
4.5	Rank correlation between workers ranked according to the raw scores attained on the <b>subsetMCQ</b> and <b>fullMCQ</b> tests, separated by choice set design strategy – choice sets optimized according to the proposed objective yield better rank correlation than choice sets sampled randomly. . . . .	75
4.6	Visualization of optimal choice sets for the question “Kansas”, optimized to students of varying prior ability parameter (black vertical bar, displayed over the empirical distribution of inferred student abilities). Observe that as ability increases, choices become clustered closer to the true answer, making the correct answer more difficult to discern. . . . .	76
4.7	Example StackExchange questions with posterior distributions over choice correctness parameters. Two optimal choices are highlighted and annotated. See Section 4.10.3 for details. . . . .	77

4.8	Example StackExchange questions with posterior distributions over choice correctness parameters. Two optimal choices are highlighted and annotated. See Section 4.10.3 for details. . . . .	78
4.9	An illustration of a two-dimensional embedding of items (e.g., movies), represented by black triangles, and users, represented by hyperplanes $\theta_1$ and $\theta_2$ . A user's relative preference towards items is represented by a projection of the item's vectors $x$ onto the user hyperplane $\theta$ , i.e., $\theta^T x$ (hollow triangles). The task of optimal choice set design is to select a cardinality-constrained subset of the items to most efficiently learn the user's preference parameters $\theta$ . . . . .	85
4.10	Accuracy in predicting users' held-out preferences as a function of the number of queries made to a user in the training session (simulation). Figures (a), (b) and (c) simulate settings where the choice sets consist of 4, 3 and 2 options respectively. Choice sets are generated (i) by solving the optimization problem in (4.8) and (ii) by sampling choices uniformly at random. Optimal choice sets outperform randomly generated choice sets, however, the gains are especially large when the number of choices shown is small. . . . .	88
5.1	A probabilistic graphical model that summarizes the proposed <i>joint assessment and grading (JAG)</i> framework. Our model jointly captures the statistical dependencies between the abilities of students that generate open-response answers ( $S_{\text{open}}$ ), the abilities of students that select the correct answers when they are presented in a form of a multiple choice test ( $S_{\text{mcq}}$ ), the underlying question difficulty $q_j$ and the observed responses ( $y_{ij}^k \in \{+1, -1\}$ ). The correctness of each open-response answer $z_j^k \in \{+1, -1\}$ (omitting the index of student that generated the answer) is a hidden variable, the state of which is inferred in addition to the remaining parameters during inference. Note that the model is able to "grade" an open-response submission of a students by integrating the response patterns of other students who were presented that open-response answer as a choices in a multiple choice version of the question. . . . .	106
5.2	Distributions used in generating synthetic data, where $p(q)$ and $p(s)$ are the distributions of question difficulty and student ability respectively. The quantity $\mathbb{E}[s - q]$ represents the average relative competency of the classroom: a large value of $\mathbb{E}[s - q]$ indicates that the majority of students will answer most of the test items correctly, and vice-versa. See Figure 5.3 for the effect of the class distribution on performance. . . . .	111

5.3	Accuracy in predicting the correct answers on synthetic data, as a function of the average relative competency in the classroom (measured in the multiples of standard deviations of the distributions). The simple majority-vote baseline performs comparably with our model for class distributions with large relative competency (since the majority of the students answer most questions correctly). The model significantly outperforms the baseline in the regime of lower relative competency (i.e., when most questions are too difficulty for the majority of the students). . . . .	112
5.4	Screenshot of a segment from each of the two Mechanical Turk tasks. Workers are required to provide an open response answer to each question in the <i>open response task</i> , and select (click) all answers that apply in the <i>multiple choice task</i> . The choices in the <i>multiple choice task</i> are aggregated from the open-response submission of other workers as part of the <i>open-response task</i> . . . .	113
5.5	Accuracy in predicting the correct answers in the dataset collected on Mechanical Turk. The model that incorporates both the <i>open-response</i> and <i>multiple choice</i> components ( <b>EM +open</b> ) significantly outperforms the model that only incorporates the multiple choice component ( <b>EM -open</b> ) and a simple majority-vote baseline. . . .	116
5.6	Accuracy in predicting the correct answers in the dataset collected on Mechanical Turk for the model initialized with the heuristic described in Section 5.4 ( <b>EM +open</b> ) and the model initialized randomly ( <b>EM +open (rand init)</b> ). Good initialization significantly improves performance, especially in the regime of little to no partially labeled data. . . . .	117
5.7	Rank correlation (kendall-tau) for students submitting open-response answers ( $S_{open}$ ) between the model-inferred ranking ( <b>EM +open</b> ) and the ranking obtained using the gold-standard correctness labels for each answer (via the Rasch model). The model generates high quality rankings with little to no labeled data, significantly outperforming the majority baseline where students are ranked using the parameters obtained from the Rasch model, but where the correctness of each answer is obtained via a majority vote). . . . .	118
5.8	Rank correlation (kendall-tau) for students submitting multiple choice answers ( $S_{mcq}$ ) between the model-inferred ranking ( <b>EM +open</b> and <b>EM -open</b> ) and the ranking obtained using the gold-standard labels for each answer (via the Rasch model). In contrast to the rank correlation for students submitting open-response answers (Figure 5.7), rank correlation for multiple choice students is lower. . . . .	119

5.9	Accuracy in predicting the correct answers in the dataset collected on Mechanical Turk as a function of the number of students answering multiple choice questions ( $ S_{mcq} $ ). More students answering multiple choice questions improves the performance of the model ( <b>EM +open</b> ) in relation to the majority baseline. . . . .	119
6.1	A high-level overview of the proposed latent choice allocation (LCA) framework: open-response submissions are combined into a multiple choice question; selections made by those students answering a multiple choice version of the question are used as “feedback” to (i) cluster similar open-response submissions and (ii) grade them, i.e., identify correct and incorrect open-response submissions. From the perspective of the students, both groups are taking a test (either open-response or multiple choice), but those answering the multiple choice questions are also implicitly grading and clustering the open-response submissions. . . . .	122
6.2	Generative process of the proposed model. . . . .	133
6.3	Example of inferred posterior distributions for two simulated configurations: 2 latent answers (top panel) and 4 latent answers (bottom panel), where question difficulty parameters $\{\beta_k\}$ are shown in red, and student ability parameters $\{s_i\}$ are shown in blue. Two pairs of latent answers in the 4 cluster configuration (bottom panel) have identical parameters (displayed with a small offset for illustration). The true and the posterior distributions over clustering configurations are displayed with a similarity matrix, where the cell $(i, j)$ in the similarity matrix represents the average fraction of MCMC samples where choice $i$ and $j$ belong to the same cluster. Additionally, explicit posteriors over the number of clusters are displayed for each of the two configurations (right-most panel). Our results show that the model is capable of recovering accurate clustering information in a challenging scenario, where two pairs of clusters have an identical set of parameters. . . . .	135
6.4	Inferred number of clusters (latent answers) vs. true number of clusters as a function of the $\epsilon$ hyper-parameter (simulation). The proposed model estimates a larger number of clusters for smaller values of $\epsilon$ . . . . .	137

6.5	Accuracy in predicting correct answers (solid black) and the average number of inferred latent answers per question (dashed red) as a function of hyper parameters $\alpha$ and $\epsilon$ (baseline accuracy is 52%). Increasing $\alpha$ leads to more clusters (latent answers) on average, while increasing $\epsilon$ leads to fewer clusters on average. Accuracy suffers when the number of inferred clusters is either too few or too many: if (i) the number of clusters is too few, the model potentially groups unrelated answers and (ii) if the number of clusters is too many (in relation to the true number of clusters), the model (incorrectly) treats multiple responses on the answers represented by the same cluster as independent observations, leading to poorer parameter estimates. . . . .	140
6.6	Answers (open-response submissions) in response to two questions from an OpenStax U.S. History textbook. Dashed lines group latent answers according to the results of the inference. Numbers next to each answer represent inferred answer correctness (answers with values $\geq 0$ can be interpreted as correct answers, and as incorrect answers otherwise, highlighted in red). A similarity (affinity) matrix (described in Section 6.3) is shown for each question. . . . .	142
6.7	Answers (open-response submissions) in response to two questions from an OpenStax Psychology textbook. See caption of Figure 6.6 (above) for details. . . . .	143
6.8	Effect of the hyper-parameter $\epsilon$ on the inferred cluster for a question from an OpenStax Psychology textbook. As expected, we observe that a larger value of $\epsilon$ results in fewer (and larger) clusters. See caption of Figure 6.6 for additional details on interpreting the figure. . . . .	144
7.1	The optimal strategy for providing a self-assessment score $\log \theta_{ij}$ for a student with ability $s_i$ on a question of difficulty $q_j$ , assuming the student's knowledge that a random fraction $\rho$ of the questions will be graded. The optimal strategy is approximately piece-wise linear as a function of the student's relative ability $s_i - q_j$ . In the regime of low relative ability, the student's optimal strategy is to report a fixed score that is a function of $\rho$ , regardless of his or her relative ability. . . . .	163

7.2	Simulation results. Rank correlation across students obtained using three models for different variance of self-grading bias ( $\sigma_2$ ): (i) <i>black</i> : a model that uses student self-scores and the correctness of their response to a subset of graded questions (number of graded questions on $x$ -axis), (ii) <i>solid gray</i> : a model that uses correctness of their response to a subset of graded questions only (number of graded questions on $x$ -axis) and (iii) <i>dashed gray</i> : a model that uses only the students' self-score. . . . .	164
7.3	Screenshot of one question from the Mechanical Turk task. A subject answers a math question and provides a self-assessment score by adjusting a slider. The student sees the number of points that they will gain if they answer the question correctly (green) and the number of points they will lose if they answer the question incorrectly (red). . . . .	165
7.4	User study results. Rank correlation across students obtained using three different models (i) <b>Self-scored</b> : a model that relies entirely on student-submitted self-assessments, (ii) <b>Graded</b> : a model that relies entirely on instructor-provided grades, as a function of the number of graded questions ( $x$ -axis), and (iii) <b>Self-scored + Graded</b> : a model that aggregates students' self-assessment scores on all questions and a variable number of instructor-graded questions ( $x$ -axis). (a) Computes rank correlation across all students using Kendall Tau, and (b) decomposes rank correlation across the first two quartiles using the <i>Precision@Quartile</i> metric. The model that combines self- and instructor-assigned scores is significantly better at predicting the top-performing students (first quartile). Combining instructor grades with self-assessment significantly improves both rank measures, especially when only a few questions are graded. Note that the total number of questions in the study was 30; we display the results up to 15, as the differences between both models is not substantial beyond that. . . . .	166
7.5	Bias vs. ability (centered). Both parameters were inferred using all of the available data. Each point in the scatter-plot corresponds to one student. A weak, but significant correlation between bias and ability exists. . . . .	167
7.6	Inferred bias parameter of each student (sorted in an increasing order). The bias parameter was inferred using all of the available data. . . . .	168

8.1	Schematic of the optimization problem for selecting words within documents to “switch”, i.e., replace their occurrence with their translation into a foreign language. In this example, pink words represent words that are “switched” ( $x_{ij} = 1$ ), while white words remain in their original language ( $x_{ij} = 0$ ), and dashed lines trace the mentions of the same word within and across documents. A gray shaded region represents a context, e.g., a sentence. . . . .	183
8.2	Screenshots of the two stages of the Mechanical Turk studies described in this chapter. The <i>reading stage</i> (a) presents readers with an article in English, with a subset of the words switched to a foreign language. The follow-up <i>quiz stage</i> presents the readers with a subset of the words they encountered during the reading, one at a time in a random order, requesting the worker to enter their guess and evaluate the precision of their guess after being revealed the meaning of the word. . . . .	184
8.3	Relationship between word (item) frequency and recall at different recall levels (higher recall levels correspond to more precise recall of the word’s meaning). Item frequency refers to the number of occurrences of the word switched to its translation in the document where it occurs. . . . .	188
8.4	Relationship between word (item) frequency and recall for different number of “switched” words per context. Item frequency refers to the number of occurrences of the word switched to its translation in the document where it occurs. . . . .	189
8.5	Proposed model for the relationship between word frequency and recall, based on the empirical findings in Figure 8.3. . . . .	191
8.6	Average number of words recalled as a function of word (item) frequency, partitioned by recall level and condition. . . . .	193
9.1	A graphical model of student learning and testing, i.e., a continuous state space Hidden Markov Model with inputs and outputs. $\mathbf{s}$ = student knowledge state, $\ell$ = lesson skill gains, $\mathbf{q}$ = lesson prerequisites, $\mathbf{a}$ = assessment requirements, and $R$ = result. . . . .	201
9.2	Geometric intuition underlying the parametrization of the assessment result likelihood (Equation 9.1). Only the length of the projection of the student’s skills $\mathbf{s}$ onto the assessment vector $\mathbf{a}$ affects the pass likelihood of that assessment, meaning only the “relevant” skills (with respect to the assessment) should determine the result. . . . .	202

9.3	The vector field of skill gains for a lesson with skill gains $\ell = (0.5, 1)$ and prerequisites $\mathbf{q} = (0.7, 0.3)$ . Contours are drawn for varying update magnitudes. A student can compensate for lack of prerequisites in one skill through excess strength in another skill, but the extent to which this trade-off is possible depends on the relative weights of the prerequisites. . . . .	204
9.4	An extremely simple embedding . . . . .	206
9.5	A two-dimensional embedding without lessons . . . . .	207
9.6	A two-dimensional embedding with lessons, without prerequisites	208
9.7	A two-dimensional embedding with lessons and prerequisites . . . . .	209
9.8	We explore the parameter space of the two-dimensional embedding with prerequisites and bias terms by doing a grid search on $(\sigma^2, \beta)$ . . . . .	212
9.9	Sensitivity of validation AUC to the “depth” of a student’s history (from $t = T - \text{depth}$ to $t = T$ ). A student’s recent history is most helpful for predicting assessment results, which we observe in the plateauing of the curve as we gradually include interactions from the students far past. . . . .	214
9.10	A schematic diagram of a <i>bubble</i> , where a group of students converges on a lesson, splits off into two different lesson sequences, then converges on the same assessment. . . . .	217
9.11	The $x$ -axis represents a threshold on absolute difference between pass rates of the two <i>bubble</i> paths. <i>Bubbles</i> are filtered to meet the following criteria: at least ten students take each branch, each branch must contain at least two lessons, and both branches must contain the same number of lessons. The error bars represent standard error, and their $x$ -coordinates are slightly perturbed so the error bars for different curves can be distinguished. . . . .	218
10.1	(a) ROC curves for the task of binary aspect classification. (b) AUC (left $y$ -axis) of aspect classification for terms with a maximum document rank given on $x$ -axis. Shaded region shows the number of terms up to the given maximum rank (right $y$ -axis). . . . .	226
10.2	Average AUC using 10 fold cross-validation at the task of binary aspect classification, applied to Chris Bishop’s <i>Pattern Recognition and Machine Learning</i> textbook. A semi-supervised approach with less than 5% of the labeled examples performs comparably to the fully supervised model with hundreds of labeled examples . . . . .	233
10.3	Average AUC using 10 fold cross-validation at the task of binary aspect classification, applied to five textbooks. Each model was trained independently for each textbook. . . . .	234

- 10.4 Posterior distribution over the units explaining *conditional independence* inside Chris Bishop’s “Pattern Recognition and Machine Learning” textbook (left is the beginning of the textbook, right is the end of the textbook). Red color the location where the index indicates the term is actually explained (gold standard annotation). 235
- 10.5 Posterior distribution over the units explaining *cross validation* inside Chris Bishop’s “Pattern Recognition and Machine Learning” textbook (left is the beginning of the textbook, right is the end of the textbook). Red color the location where the index indicates the term is actually explained (gold standard annotation). . . . . 235
- 10.6 Each document is represented by a blue shaded region: the top part corresponds to the explained set  $E_i$  and the bottom part corresponds to the assumed set  $A_i$ . Red dots correspond to terms. This is an example of a feasible solution, where each document is *covered*. . . . . 237
- 10.7 Optimal solution for a small toy problem obtained with an ILP. Each row is a unique term and each column is a document corresponding to a position in a sequence (from left to right). A blue square indicates that the term is explained in the document, and a red square that the term is assumed. Notice that all of the constraints are satisfied: each term is explained in some document before it is assumed and each position in the sequence is occupied by a single document. . . . . 243
- 10.8 Term aspect classification is useful at the task of recovering prerequisites for units within a textbook. The  $y$ -axis is the average AUC at the task of predicting whether a particular unit is a prerequisite of another unit, based on three metrics. The metric that incorporates the *Explain/Assume* classifier performs best (solid line). 249
- 10.9 An example of two different web-pages about the same topic: *Expectation Maximization*, together with each page’s terminology and its classification into either the *Explained* class (green) or the *Assumed* class (red). Observe that the two pages, while about the same topic, are different in what they assume about the reader. The article on the left is a very basic introduction to this topic, while the article on the right is written for experts. . . . . 256
- 10.10 An example optimal sequence for the target document on *Maximum Likelihood Estimation*. Left: the term-coverage diagram. Each column represents a single web-page and each row a single term. Red rectangles correspond to terms that are classified as *assumed* in the corresponding document and *blue* corresponds to the *explained terms*. Right: the term-cover diagram is converted into a directed graph whereby an edge is drawn to a document from its closest prerequisite that explains at least one assumed term. . . . 257

10.11	Additional examples of optimal paths generated from the 1,000-document web-page corpus for a select set of target web-pages. See text for details. . . . .	257
10.12	A high-level view of a 42,000 document cross-section of the web in the areas of Machine Learning and Statistics. Each node in this graph represents a cluster of web-pages obtained using the same keyword (displayed) to Bing Search API. Fundamental “source nodes” with high out-degree (blue) represent web-pages that explain concepts assumed in many other pages; complex “sink nodes” with high in-degree (orange) represent web-pages that assume concepts explained in many others. . . . .	258
10.13	An example sequence of documents (columns) that lead the user to the goal document on the topic of <i>Gibbs Sampling</i> . . . . .	260

## CHAPTER 1

### INTRODUCTION

The Internet, with its potential to instantly reach every corner of the world, has recently demonstrated tremendous implications for the world of education. Online platforms that deliver high quality educational content to learners across wide social, economic and ethnic groups, carry the promise of bridging the equity gap that is so pervasive in the education systems today.

Providing equality in access to the same educational resources, however, while an important first step, does not immediately result in *educational equity* — an idea that learners of different backgrounds require different resources in order to achieve the same outcomes. This distinction between educational equality and equity is amplified as Internet classrooms further extend their reach and invariably subsume underserved and disadvantaged communities around the world. In addition to providing equality in access, achieving equity requires recognizing and catering to the growing diversity of the learners by (i) assessing learners' background and (ii) delivering content to learners that suites their background and learning needs. These two steps, when repeated continuously, form the basis of *mastery learning* — a learning model originally credited to Bloom [20, 32] — whose key principle is a feedback loop that continuously evaluates the learner's current knowledge and then provides corrective feedback and the necessary learning material to ensure consistent growth in mastery.

Over the course of almost four decades since the introduction of this concept, mastery learning has been subjected to rigorous evaluations across different levels of instruction, with the consensus on its effectiveness in improving learning outcomes. On the basis of these four decades of inquiry into mastery learning,

Benjamin Bloom, one of the founders of mastery learning, concluded:

*“What any person in the world can learn, almost all persons can learn if provided with appropriate prior and current conditions of learning.”*

(Benjamin Bloom)

Implementing mastery learning in a classroom, however, has proved to be a highly labor-intensive process, as it burdens the instructor with the task of tailoring their instruction to the individual students’ backgrounds and needs. Understandably, the rapid proliferation of computing technology in the 70s and 80s had rejuvenated the interest in practical mastery learning, which to a great extent served as a foundation in the development of the field of Intelligent tutoring systems (ITS).

Since then, over three decades of research and fielding of Intelligent tutoring systems in isolated domains, such as math and science tutoring, had shown promise in automating components of the mastery loop. At the same time, it also highlighted their key limitation: *content*. As Intelligent tutoring systems are but an algorithmic layer over static content (e.g., assessment and learning material), their performance is fundamentally limited by the extent and quality of the content over which they operate. The responsibility of content curation falls on the instructor, who ultimately presents a bottleneck in the scalability of all ITSs. And while scalability was not a primary concern for early ITSs that were designed for and evaluated in traditional classrooms, this limitation becomes amplified as classrooms scale themselves to the web, e.g., via Massively open online courses (MOOCs) and public learning platforms like Khan Academy. In catering to the diverse learners, the value of diversifying content becomes as (if not more) important as the algorithms for organizing that content.

The thesis of this dissertation is that *scalable learning tools must simultaneously address the tasks of automatic content generation and curation*. We further posit that scalable content generation and curation in the near term can be achieved by a combination of automation and crowdsourcing by leveraging the learners themselves within the loop of content generation. We follow the natural structure of mastery learning in organizing this dissertation: in the first part (chapters 3 to 7) we develop models for scalable assessment, and in the second part of the dissertation (chapters 8 to 10) we focus on developing models for learning content generation and curation. The rest of this chapter briefly outlines the key contributions of both parts of this dissertation.

## **1.1 Contributions of this dissertation**

### **1.1.1 Assessment**

Assessment plays a key role in diagnosing the strengths and deficiencies of individual learners in the context of the mastery loop. In addition to providing feedback to the instructors, assessment content in the form of questions provides learners with an opportunity to reflect on their own shortcomings in mastery. In the first part of this thesis, we will develop and evaluate a number of *assessment frameworks* that aim to assist both learners and instructors in assessing mastery. By *assessment framework*, we refer to both — (i) the process by which assessment content (instrument) is generated (e.g., a question) and (ii) the process by which assessment of students' mastery is performed (e.g., how students are evaluated using the assessment instrument). For the purpose of framing our contributions,

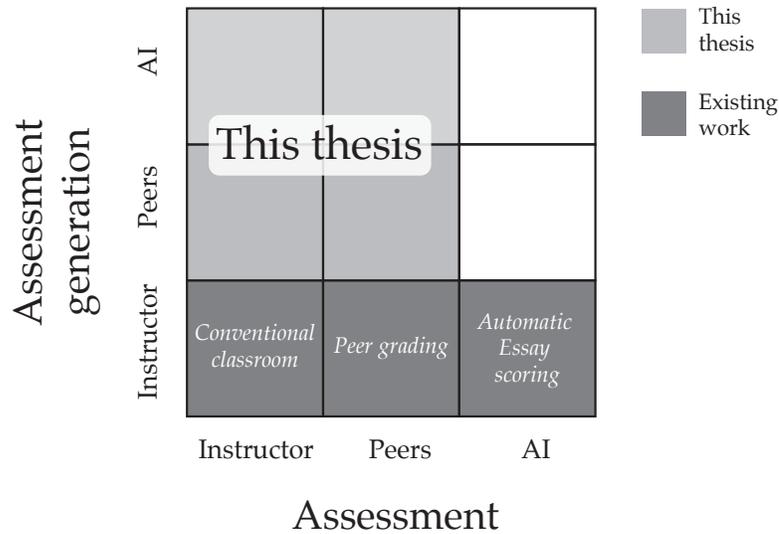


Figure 1.1: A space of all *assessment frameworks* (which broadly encompass both, the process for generating assessment content and administering it), can be represented in terms of the roles delegated to the instructors, peers and automation. For example, traditional tests generated and graded by instructors would be at the lower left corner. The contribution of the first part of this thesis lies in methods for scalable assessment *generation* and *administration*.

it would be useful to broadly generalize any such assessment framework as belonging to a space of assessment frameworks, defined by the roles delegated to instructors, peers and automation in both — the process of generating an assessment instrument and the process of administering assessment (i.e., evaluating learners using the assessment instrument) (see Figure 1.1). For example, conventional questions created and graded by an instructor would be at the lower left corner, while a hypothetical framework that automatically generates and grades questions would be at the top right.

In this thesis, we explore a spectrum of assessment frameworks that vary in the extent to which they rely on peers, instructors and automation in *both* — assessment generation and assessment administration. We briefly outline the

motivation and contribution of each chapter.

### **Automatic question generation**

In this chapter we demonstrate the potential of fully automatic question-generation from the web, with the focus on its potential in aiding learners in their self-evaluation of mastery during learning. Our key contribution is a scalable approach for broad-coverage, high-level question generation from informational texts (e.g., encyclopedic articles) that encourage learners to reflect on the depth of their understanding of the material.

### **Crowdsourcing assessment**

A significant portion of our contribution to scaling assessment, lies in the concept of *crowdsourcing assessment*, which broadly encompasses methods for delegating all or fraction of the task of assessment generation (e.g., creating questions) and assessment administration (e.g., evaluating learners using these questions) to learners themselves. The advantages offered by crowdsourcing assessment instruments are (i) leveraging the size of web-scale classrooms allows for scalable and diverse assessment generation that has the potential to readily adapt to the dynamics of the curriculum (ii) learner-driven question design carries the potential to naturally reflect the learners' strengths and deficiencies, thus creating assessment instruments that probe relevant misconceptions in a targeted population of learners. Our approach for crowdsourcing assessment consists of aggregating students' open-ended answers into multiple choice questions, facilitating scalable grading, while simultaneously generating questions that can

be adapted to learners of different levels of mastery. The three problems that arise naturally in the task of crowdsourcing multiple choice questions are: (i) how to generate choice sets (distractors) that are optimal (e.g., most informative of mastery) for a target population of learners, (ii) how to assess learners (e.g., rank students by their mastery) using crowdsourced questions and (iii) how to identify groups of “common errors” among students’ submissions in order to present questions consisting of only the representative errors, as well as provide an instructor with a high-level overview of the students’ mastery of the material. We briefly describe our contribution in addressing each of these three problems of crowdsourcing assessment.

- **Optimal multiple choice questions:** In Chapter 4, we formulate the problem of generating *adaptable multiple choice questions*, i.e., multiple choice questions whose difficulty can be “tuned” by optimizing their choice-set (or option-set), with the goal of generating questions that can be automatically adapted to learners of different mastery level.
- **Joint assessment and grading:** In chapters 5 and 6, we develop a methodology for assessing learners using crowdsourced multiple choice questions, where the choices are comprised of open-ended answers submitted by other learners. By transforming open-ended submissions into questions, our assessment framework makes the process of grading open-ended submissions implicit. The proposed framework of *Joint assessment and grading* or JAG, solves several important problems in assessment: (i) incentivizing students’ effort in peer-grading is made implicit by re-framing grading as assessment, (ii) multiple choice questions composed of open-ended submissions from the same population of learners naturally reflect common errors and misconceptions (a key feature of well-designed multiple choice

questions and a difficult task in manual question design) and (iii) by leveraging students' answers to multiple choice questions, our framework is able to automatically group semantically similar open-ended submissions into clusters, thus amplifying the scalability in open-response grading. We demonstrate that our framework is able to perform summative assessment of learners (i.e., rank learners according to their mastery) with minimal instructor input.

- **Self-assessment:** In chapter 7, we develop an approach for *calibrated self-assessment* — a highly scalable method for summative assessment that relies on students grading themselves. Self-assessment aids learners in self-reflecting on own mastery, while at the same time providing instructors with a calibrated measurement of the mastery of the entire classroom. Grade calibration is achieved via a statistical model that compensates for the individuals' biases in self-evaluation by relying on a small set of instructor- and/or peer- graded assignments.

### 1.1.2 Learning content curation

In addressing the second component of the mastery learning loop — learning content generation and sequencing — we develop and evaluate three data-driven models that leverage the scale of the web in order to curate learning experiences: (i) language learning from reading the web, (ii) learning content representation from students' interaction logs (access traces) and personalized lesson sequencing and (iii) learning from heterogeneous learning resources on the web. We briefly summarize our contribution to each problem below.

## **Language learning from the web**

Research into language acquisition has demonstrated the effectiveness of *incidental vocabulary learning* through reading — an idea that learners can acquire the meaning of new words through context in which these words appear. With increasingly more reading done by users on electronic devices and on the web, we propose to optimize the presented web content such as to maximize the effectiveness of foreign vocabulary acquisition from context. In chapter 8, we build on the theory and experimental evidence of incidental vocabulary learning, in order to develop and evaluate an optimization framework for presenting web pages to aid in incidental vocabulary acquisition.

## **Learning content representation and personalized lesson sequencing**

In chapter 9, we address the problem of learning from students' interactions with the content, in order to automatically (i) identify the educational value of the different pieces of learning material, and (ii) predict effective learning paths for learners of different ability. We develop a model that learns a joint representation of students, assessment and learning content from traces of students interactions (e.g., logs consisting of the questions that students answered, and reading material that students completed), and use that representation to evaluate its potential effectiveness in personalizing lesson sequences. We develop a technique for debiasing logged data of students' interactions in order to evaluate the proposed representation in its potential to recommend effective learning paths.

## Web-scale learning

In chapter 10, we propose to address the “content bottleneck” in content personalization, by leveraging the entirety of the web and its diversity, as a source of learning material for personalization. A growing subset of the web today is aimed at *teaching* and *explaining* technical concepts with varying degrees of detail and to a broad range of target audiences. Content such as tutorials, blog articles and lecture notes is becoming more prevalent in many technical disciplines and provides up-to-date technical coverage with widely different levels of prerequisite assumptions on the part of the reader. We propose a task of organizing heterogeneous educational resources on the web into a structure akin to a textbook or a course, allowing the learner to navigate a sequence of web-pages that take them from point A (their prior knowledge) to point B (material they want to learn). We approach this task by (i) performing a shallow term-level classification of what concepts are *explained* and *assumed* in any given text, and (ii) using this representation to connect web resources that explain concepts to those web resources where the same concepts are assumed. Our main contributions are (i) a supervised and a semi-supervised approach to identifying explained and assumed terms in a document and (ii) an algorithm for finding optimal paths through the web’s learning resources given the constraints of the user’s goal and prior knowledge.

## 1.2 Summary of contributions by research collaborators

In this section, we outline and acknowledge contributions of research collaborators to this thesis:

**Chapter 3:** This work was initiated during an internship at Microsoft Research in 2014 and was supervised by Sumit Basu<sup>1</sup> and Lucy Vanderwende<sup>2</sup>. The majority of the introduction section of this chapter was written by Sumit and Lucy.

**Chapter 4:** The ILP optimization algorithm (Section 4.6), its analysis and proof of correctness were contributed by Frans Schalekamp<sup>3</sup> who was a visiting professor in the Operations Research department at Cornell during 2015. Claim 4 and its proof in Section 4.13.3 is also due to Frans. We thank Kelvin Luu for being involved in the early stage discussions of ideas behind this work.

**Chapter 9:** The work described in this chapter was a close collaboration with Siddharth Reddy<sup>4</sup>. The idea of learning an embedding from students' interactions was due to Sid. The graphical model was developed jointly. The synthetic and "bubble" experiments were primarily devised by Igor Labutov. All code and the majority of the text was written by Sid, with the exception of the portion of Section 9.6.2 describing the "bubble" experiments (and the introduction and conclusion). The data was obtained by Sid as part of an internship with Knewton<sup>5</sup>, a learning technology company. The project was supervised by Thorsten Joachims.

**Chapter 10:** The NP-hardness and approximation-hardness proofs in Section 10.5 respectively were contributed by Frans Schalekamp.

---

<sup>1</sup><http://research.microsoft.com/en-us/um/people/sumitb/>

<sup>2</sup><https://www.microsoft.com/en-us/research/people/lucyv/>

<sup>3</sup><http://fransschalekamp.com/research/>

<sup>4</sup><http://siddharth.io/>

<sup>5</sup><https://www.knewton.com/>

## CHAPTER 2

### BACKGROUND

In this chapter, we frame our contributions in the context of a long and rich history of research into diverse areas that we broadly group under a general term of *educational data mining* that encompasses (i) *psychometrics*, (ii) *intelligent tutoring systems* and a relatively new direction of *learning at scale*. Following the structure of this dissertation, we partition our literature review according to the two components of the mastery learning loop: *assessment* and *learning content curation*. In addition to the general overview of relevant literature in this chapter, we include additional chapter-specific references within the respective chapters.

#### 2.1 Assessment

Development of instruments for objective measurement of psychological traits, such as intelligence, has traditionally been the focus of an area of *psychometrics* [87]. Broadly, psychometricians focus on the design of assessment instruments (e.g., test questions, questionnaires) and the methods for performing assessment with these instruments that guarantee validity (i.e., instrument measures what it intends to measure) and reliability of measurement (i.e., instrument is consistent) [87]. A cornerstone of psychometrics is the Item response theory (IRT) [53] — a set of statistical models for estimating learners’ latent traits, such as intelligence, on the basis of the items (questions) that they answer and the correctness of their responses to these items. In what follows, we review the key aspects of the classic Item response theory, as well as its more recent extensions that leverage the rich datasets afforded by the introduction of technology into the classrooms.

### 2.1.1 Psychometrics

The simplest of IRT models, the Rasch model [149], posits that the likelihood of a learner correctly answering a question is a function of that learner's ability and the question's difficulty. The classic Rasch model yields a one-dimensional embedding of items and students within a shared space that facilitates a fair comparison of learners' mastery, regardless of the items that each student answers. As a result, Rasch-like models form the foundation for adaptive testing, where students, by design, answer different sets of questions, depending on their ability. Various extensions of the Rasch model have been proposed since its initial introduction almost sixty years ago (1960), focused on extending the model to account for additional components of the variance, e.g., incorporating item-specific "discriminability" parameter to account for items that may not be informative [110] and extending item and user representation to multiple dimensions in order to account for independent skills/knowledge components [113, 194].

More recent extensions of IRT models take advantage of the the rich data that becomes available in classrooms where many of the interactions between students and content are digitally logged. Such interaction *traces*, in addition to including items (questions) that students answer and the outcomes of their responses (e.g., correct/incorrect), also incorporate learning content that students interact with (e.g., lecture notes, lecture videos, simulations) over the duration of the course. Extensions of the classic IRT models to modeling the temporal changes in students' ability, e.g., as a consequence of learning new material, include time-varying IRT and its variants [47, 180, 101]. An important result from models that incorporate the effect of lessons is that variability of students'

“paths” through learning content is critical for the model to effectively estimate lesson-specific gains.

### **2.1.2 Adaptive testing**

Item response theory forms the statistical foundation for adaptive testing – a notion that a more reliable estimate of students’ mastery may be obtained when questions are tailored to the individual students based on their responses to the previous questions. As IRT models (such as the Rasch model) are probabilistic in nature (i.e., they specify the likelihood of observing a response outcome conditioned on the latent question and student traits), they can be naturally framed in the context of the optimal experiment design paradigm [51], i.e., finding the next most informative question of the student’s mastery given the model’s current estimate of the student’s ability. Adaptive testing plays a core role in the context of the mastery learning loop, where because of the inherent feedback between assessment and instruction, more effective testing implies more effective instruction.

Adaptive testing has been successfully deployed in high-stakes tests such as GMAT (Graduate management adaptive test) and GRE (Graduate record examination) and first studies of adaptive testing date back to the 1980s, demonstrating that adaptive testing has the potential to increase precision in proficiency estimation while reducing the number of questions [196, 195]. All IRT-based adaptive testing mechanisms rely on the notion of minimizing variance of the estimated quantities (i.e., student ability) from the set of presented question (e.g., via maximizing Fisher information in maximum likelihood setting or prior-posterior gain

in a Bayesian setting). While most adaptive assessment systems focus on the task of selecting only the next question (online design), some attention has also been given to the task of test-bank design [53] (batch design), where the optimization objective is similar to the online setting, with the exception that no feedback is obtained from the student until all items in the batch have been answered.

At its core, adaptive testing is an application of optimal experiment design to the task of estimating learner proficiency. Optimal experiment design primarily concerns with the task of identifying the most effective measurement to take, in order to minimize estimator variance of some unobserved parameter (e.g., student proficiency in the adaptive learning setting). The field of optimal experiment design has a rich history, dating back to work of Peirce (optimal experiment design in regression) and Fisher, with one of the early successful application dating back to the late 19th and early 20th century in the application to agricultural and geospatial measurements [141, 54, 55]. Optimal experiment design has been applied to a number of probabilistic models and domains, e.g., gaussian processes in spatial sensing [213, 94], dynamic system identification [210, 159] and medical imaging [145].

### **2.1.3 Learning at scale**

Massive open online courses (MOOCs) and online learning platforms such as Khan Academy have generated a new challenge in assessment — performing scalable assessment of thousands of students. As massive classrooms imply massive student-to-teacher ratios, scalability in assessment can be maximized by minimizing instructor effort. Three methods have been adopted by MOOC

practitioners in addressing the issue of assessment scalability: *multiple choice testing*, *peer grading* and *submission clustering*. We review these methods in the order of their prevalence in today's MOOCs.

### **Multiple choice testing**

Although the tool of *multiple choice testing* is now nearly one hundred years old, it remains the workhorse of modern MOOCs. The model of multiple choice testing, introduced by Benjamin Wood [45], one of the fathers of educational psychology, remains appealing for its scalability, as multiple choice questions by design require no manual effort in grading. Despite its scalability, there are several reasons for why multiple choice testing alone may be insufficient in classroom assessment. In education research, open-ended questions that encourage learners to think freely, have been shown to facilitate higher-order thinking and greater engagement, in contrast to multiple choice questions [67]. Multiple choice question design has also received significant attention, focusing on problems such as ensuring validity [68, 67], selecting an optimal number of distractors [157, 69] and the methodology for selecting effective distractors [70, 67, 186]. Critically, ensuring that constructed multiple choice questions present effective assessment instruments requires instructors to be attuned to the misconceptions and common errors in the specific domain and the population of students being tested [68] — an especially daunting task in classrooms that are scaling to different regions of the world. Both limitations of multiple choice questions were well articulated by one of the pioneers of machine teaching, B. F. Skinner in his 1960 paper titled *Teaching Machines* [179]:

*“The student must compose his response rather than select it from a set of alternatives, as in a multiple-choice self-rater. One reason for this is that we want him to recall rather than recognize — to make a response as well as see that it is right. Another reason is that effective multiple-choice material must contain plausible wrong responses, which are out of place in the delicate process of ‘shaping’ behavior because they strengthen unwanted forms. Although it is much easier to build a machine to score multiple-choice answers than to evaluate a composed response, the technical advantage is outweighed by these and other considerations.”* (B. F. Skinner)

To address this challenge, domain-specific methods for automatically generating multiple choice questions have been proposed [26, 12, 128]. In chapters 4, 5, 6 we propose and evaluate several domain-agnostic methods for generating multiple choice questions based on students’ open-response answers.

### **Peer grading**

Recognizing the limitations of designing and administering multiple choice questions, *peer-grading* has received much attention as an alternative approach to scalable assessment that facilitates grading of open-response answers, e.g., short answers, programming assignments, projects and essays. The principle behind peer grading is the delegation of the grader role to some or all of the students in the class, thus potentially multiplying the effort of a single instructor [95]. The resulting increase in grading throughput, however, is partially offset by the variance in the reliability and bias of students’ grading ability, and requires statistical machinery for aggregating peer-submitted grades. The interest of the machine learning community in the task of peer-grading has been primarily

motivated by the problem of *grade aggregation*, i.e., developing optimal or near-optimal techniques for aggregating multiple grades from “noisy” graders into a final grade to be assigned to the student. Methods such as [143] and [148] explicitly model student’s “grading ability” via inherent bias and reliability and effectively estimate the final grade (or rank as in [148]) by appropriately weighing individual grader’s contributions in relation to the grader’s “ability to grade”.

### **Submission clustering**

Another direction of research into scalable assessment is *submission clustering* – the process of grouping related submissions into clusters and requiring instructors to grade only the representative solutions from each cluster, effectively amplifying instructors’ efforts. While recent research [174] indicates that submission clustering is an effective solution to scalability in assessment, the task of automatically clustering similar submissions is difficult. Domain-specific solutions for submission have been proposed [9, 64, 130, 24, 106, 86, 207, 60, 185]. In chapter 6, we propose a domain-agnostic approach for submission clustering and grading, by transforming students’ solutions into multiple choice questions, and relying on the response pattern of students answering the multiple choice questions, to automatically identify groups of semantically related submissions.

### **2.1.4 Crowdsourcing**

Peer-grading can be viewed as a special-case of a more general problem, known as *label aggregation* or rank aggregation in crowdsourcing. Both tasks can be seen as instances of aggregating judgement from the crowd with the goal of producing

a more reliable estimate of some underlying truth – a notion of the *wisdom of the crowd* that dates back to de Caritat in the 18th century [43]. De Caritat had observed that in a population where each person makes an independent guess as to the truth of some underlying but unobserved quantity with probability of being correct greater than  $1/2$ , the aggregate will invariably converge to the true value (in the limit of an infinite population) [43]. In the early 20th century, Sir Francis Galton had first demonstrated experimentally that aggregating human judgements can yield aggregate estimates that are extremely close to the true value (in his experiment, the task was estimating the weight of an animal) [58]. Since the introduction of computing, more sophisticated methods have been proposed for the task of judgement aggregation, e.g., methods that explicitly learn the proficiencies of individual members of the crowd (i.e., abilities) and/or difficulties of the tasks [42, 199, 7, 204, 85, 148, 61, 147, 198, 76].

### **Learnersourcing**

A relevant recent direction of research is *learnersourcing*, i.e., relying on learners for crowdsourcing portions of the course content. A prominent example of *learnersourcing* is work by [200], where the authors rely on students to write their own explanations for how they obtained solutions to the assigned problems. Their proposed system balances exploration and exploitation to simultaneously find and present effective explanations to learners. Another example of learnersourcing is Crowdy [90], a system for soliciting learners to submit in-video-lecture summaries of what they learned, and then presenting learners with multiple version of other students' summaries, in order to then identify the best summaries from students' feedback. A similar approach was recently taken to administer

opinion polls on Wikipedia [163] — where a system allows users to submit free-response opinions, and other users then vote on a set of submitted opinions presented in the format of a poll. Our work presented in chapters 4, 5, 6 shares the feature of combining user-submitted content in the form of choices, but further leverages that signal for tasks such as assessment, optimal test design and submission clustering.

## 2.2 Learning content

In this section, we focus on the instruction component of the mastery learning loop, i.e., given our assessment of students' learning, what content do we show to the student and when? Attempts at automating this task via *machine teaching* or machine instruction follow a rich history, dating back to the early 20th century with the pioneering work of Sidney Pressey, and later B.F. Skinner. While these early machines were mechanical in nature and rigid in their capabilities, they for the first time demonstrated the potential of scaling mastery learning via automated, albeit limited, personalized instruction. With the proliferation of digital computers in the 60s and 70s, machine teaching had seen a second wave of developments, establishing the fields of Computer assisted instruction (CAI) and later Intelligent tutoring systems (ITS) whose aim was to leverage digital computers to provide greater and more effective personalization in a wide variety of domains. The proliferation of the web in the 1990s had brought another dimension to computer-based instruction, with the focus on designing hypermedia-rich and interactive instructional content (e.g., web-based textbooks). We are now at the threshold of another major development in computer-based instruction, prompted by the growing availability and speed of Internet access. With the rise

of Internet-based classrooms, facilitating real-time interactivity among learners and instructors, the challenge of machine teaching in the next decade will be in developing ways to cater instruction to the growing diversity in world-scale classrooms. In the following sections, we trace some of the key developments of machine teaching, from its roots in the early 20th century, to today.

### **2.2.1 Early Intelligent tutoring systems (1920s-1990s)**

In 1912, Edward Thorndike, one of the fathers of educational psychology, speculated on the role of a machine in the future of education:

*“If, by a miracle of mechanical ingenuity, a book could be so arranged that only to him who had done what was directed on page one would page two become visible, and so on, much that now requires personal instruction could be managed by print. (p. 165) ”* (Edward Thorndike)

The first teaching machine was a mechanical device invented by Sidney Pressey in the early 1920s [115]. The main purpose of Pressey’s machine, however, was testing: the learner was presented with a multiple choice question through a window in the machine, which he or she answered by pressing a button corresponding to the choice. The learner was then provided with immediate feedback, and if they were correct, the next question was displayed. Because the learner would not proceed to the next question until the current question was answered correctly, one could argue that the machine was implementing a rudimentary form of mastery teaching, where given a suitable sequence of questions that progress in difficulty, the learner is forced to progress through the

material at their own pace, moving to the more advanced material only when the prerequisite material has been sufficiently mastered.

While Pressey's machine was, to some extent, capable of teaching implicitly through a combination of assessment and immediate feedback, assessment remained its primary function. In 1950s, B. F. Skinner, an education psychologist, pioneered the development of more advanced machines designed explicitly for mastery instruction. The key principle that distinguished Skinner's machines was their explicit programming to facilitate incremental learning, with a mechanism for repetition of material that the student failed to master. B. F. Skinner describes the key principle behind his machine as follows:

*"Once the teacher had decided [ what he wants the student to do by the end of the term ], he has his end points — complete ignorance and complete competence. He must now bridge them with a series of eight or ten thousand steps — not an easy task. "*

(B. F. Skinner)

Skinner's machines were adopted and experimented with in schools and colleges [115]. Their key characteristic was the ability to adapt the rate of instruction to slow and fast students alike — the key characteristic of mastery instruction.

With the advent of digital computers in the 1950s, their potential in automating instruction was quickly realized. One of the first applications of digital computers (IBM mainframe) was in teaching binary arithmetic [115, 151, 152], ushering the field of Computer-assisted instruction (CAI). One of the distinguishing characteristics of the early digital programs in instruction was the ability of the questions and content presentation to adapt to the performance of the user (e.g., a more difficult question could appear to a learner who is consistently

correct). This was a significant step forward over the mechanical systems of Pressey and Skinner, where the order of the presented content was fairly rigid, allowing only for a limited repetition of the existing content.

Various CAI systems leveraging the flexibility of digital computers were developed in the following decades. One of the key systems developed during the 60s was TICCIT (Time-Shared Interactive Computer Controlled Information Television). TICCIT was the deployed for freshman level university math and English and was the first system formally evaluated by ETS, with findings concluding that computer instruction resulted in significant gains over traditional instruction, though many of the students still preferred traditional lectures [123].

PLATO (Programmed Logic for Automatic Teaching Operations), was the first general purpose CAI system developed in the 60s, designed to allow instructors to develop their own instructional material via a specialized language called TUTOR [123]. PLATO evolved into the 80s as remote mainframe terminals became more commonplace in the classrooms, allowing hundreds of students to connect to the PLATO mainframe simultaneously, fueling its wide adoption. The TUTOR programming language, the underlying mechanism for organizing lessons and assessments, was advanced in its flexibility to parse variations in students answers, and in allowing instructors to specify complex sequences and conditionals in personalizing students' paths through the content based on their performance [176]. At the same time, the potentially daunting task of organizing the structure of the course (accounting to variations based on differences in students' responses) was delegated entirely to the course instructor.

## 2.2.2 Modern Intelligent tutoring systems (1990s-2016)

In the late 1980s, and through the 1990s, the field of Computer-assisted instruction had gradually shifted focus towards student modelling, with the goal of developing intelligent algorithms for diagnosing students' deficiencies and correcting them through automatic lesson sequencing and problem generation (the tasks that had to be manually programmed in earlier systems such as PLATO). With the shift towards learner-modeling, the field of Computer-assisted instruction became also known as the field of Intelligent tutoring systems (ITS), reflecting the goal of automated tutoring, much aligned with the concept of mastery learning.

The foundation for many of the successful ITS systems in the late 90s to today is the Cognitive tutor (CT) framework [39, 92]. A cognitive tutor models procedural knowledge of a student (e.g., steps in adding fractions or long division) as skills (rules) that the student combines to solve a problem [44]. The task of a cognitive tutor is to infer the state of the students' skills by observing their responses to the problems, via a process known as Knowledge tracing (KT) [39]. Knowledge tracing observes students' solutions and estimates the probability that a student had mastered a particular skill, e.g., in programming [39] and algebra tutoring systems [92]. Since its introduction in [39], knowledge tracing had received significant attention from the psychology, education and machine learning communities, with the goal of improving the student model and its performance in estimating students' hidden knowledge state from observations. Bayesian knowledge tracing (BKT) [40] naturally captures the model's uncertainty over learners' skills via a Hidden Markov Model [146]. Recently proposed Deep knowledge tracing (DKT) [144] utilizes a recurrent neural network to model students' skill level without requiring an expert annotation of problem-specific

skills. Various extensions of Bayesian knowledge tracing have been proposed, e.g., to include the effects of lessons [65, 137, 136]. A large-scale study in Pittsburgh's high-schools at the task of algebra tutoring, has shown BKT to be an effective learning model in cognitive tutoring [92].

### 2.2.3 Hypermedia

Another direction in intelligent tutoring, is that of automatic content sequencing or planning. It differs from cognitive tutoring described above, in that while cognitive tutoring attempts to diagnose specific procedural problem-solving skills and re-mediate them, content sequencing is a broader task aimed at providing the most efficient path given the learner's background and their performance on assessment items [44]. This task can be traced back to the 1980s, with the pioneering work of automatic curriculum planning of [119, 140].

With the proliferation of the web in the mid 1990s, the research into automatic curriculum planning and content sequencing had received significant attention. The introduction of hypermedia — rich, structured content that facilitates flexible navigation — led to a stream of research in the development of adaptive hypermedia-based textbooks that adapt to the learners' knowledge and goals. Pioneering work in the area of adaptive hypermedia textbooks is due to Peter Brusilovsky, with the introduction of the InterBook — a hypermedia textbook authoring tool that allows for an explicit annotation of concepts and their relationships, as well provides an adaptive user interface that recommends personalized sequences of content in response to the students' performance (i.e., quiz performance) [119, 28]. The domain and student models in the context

of such hypermedia textbooks are authored manually, and the sophistication of both models have been extended to incorporate a variety of content (e.g., presentations, exercises, multiple choice questions) and a diversity of relations (e.g., strength of a prerequisite relation between units) [208], as well as the methods for planning individualized sequences through the content [27]. A more sophisticated learner model was developed by [4] for the task of generating more effective hypermedia links within textbooks that accommodate students' learning preferences. Hypermedia based textbooks have been shown to increase engagement and learning outcomes [27, 4]. Our work in chapters 10 takes the natural next step in developing adaptive learning content by leveraging the the open web as the source of educational content, alleviating the need for manual content curation and annotation. The need for open-content hypermedia instruction from the web is recognized and studied in [29, 108], who perform a number of preliminary studies of different techniques for automatically annotating novel content with concepts, and the potential of these techniques for developing intelligent tutoring platforms based on open content.

In the next and the following three chapters, we begin with the focus on assessment, and then shift our focus to learning content curation for the remainder of the dissertation.

## CHAPTER 3

### QUESTION GENERATION FROM TEXT

#### 3.1 Introduction

In this chapter, we focus on the task of automatic question generation from text — an integral component of assessment. The key contribution of the work described in this chapter is an approach for generating deep (i.e., high-level) comprehension questions from novel text that bypasses the myriad challenges of creating a full semantic representation.

Questions are a fundamental tool for teachers in assessing the understanding of their students. Writing good questions, though, is hard work, and harder still when the questions need to be deep (i.e., high-level) rather than factoid-oriented. These deep questions are the sort of open-ended queries that require deep thinking and recall rather than a rote response, that span significant amounts of content rather than a single sentence. Unsurprisingly, it is these deep questions that have the greatest educational value [5, 6, 122]. They are thus a key assessment mechanism for a spectrum of online educational options, from MOOCs to interactive tutoring systems. As such, the problem of automatic question generation has long been of interest to the online education community [129, 171], both as a means of providing self-assessments directly to students and as a tool to help teachers with question authoring. Much work to date has focused on questions based on a single sentence of the text [11, 109, 118], and the ideal of creating deep, conceptual questions has remained elusive. In this chapter, we hope to take a significant step towards this challenge by approaching the problem in a

---

<sup>0</sup>This chapter has been adapted from the paper [97]

somewhat unconventional way.

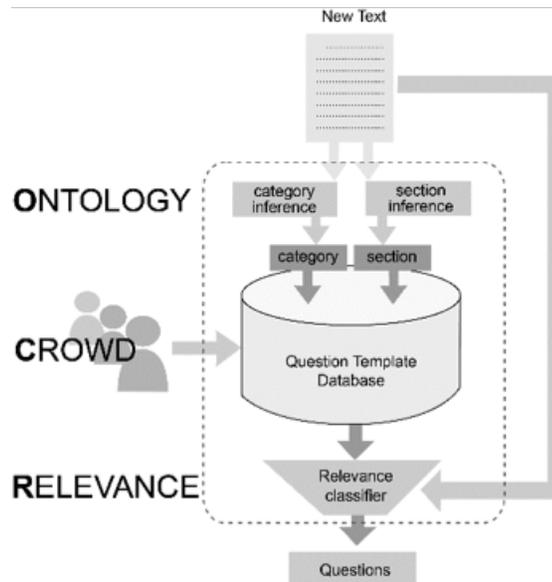


Figure 3.1: Overview of our ontology-crowd-relevance approach.

While one might expect the natural path to generating deep questions to involve first extracting a semantic representation of the entire text, the state-of-the-art in this area is at too early a stage to achieve such a representation effectively. Rather we take a step back from full understanding, and instead propose an ontology-crowd-relevance workflow for generating high-level questions, shown in Figure 3.1. This involves (i) decomposing a text into a meaningful, intermediate, low-dimensional ontology, (ii) soliciting high-level templates from the crowd, aligned with this intermediate representation, and (iii) for a target text segment, retrieving a subset of the collected templates based on its ontological categories and then ranking these questions by estimating the relevance of each to the text at hand. In this chapter, we apply the proposed workflow to the Wikipedia corpus. For our ontology, we use a Cartesian product of article categories (derived from Freebase) and article section names (directly from

Wikipedia) as the intermediate representation (e.g., category: Person, section: Early life), henceforth referred to as category-section pairs. We use these pairs to prompt our crowd workers to create relevant templates; for instance, (Person, Early Life) might lead a worker to generate the question “Who were the key influences on <Person> in their childhood?”, a good example of the sort of deep question that can’t be answered from a single sentence in the article. We also develop classifiers for inferring these categories when explicit or matching labels are not available. Given a database of such category-section-specific question templates, we then train a binary classifier that can estimate the relevance of each to a new document. We hypothesize that the resulting ranked questions will be both high-level and relevant, without requiring full machine understanding of the text in other words, deep questions without deep understanding. In the sections that follow, we detail the various components of this method and describe the experiments showing their efficacy at generating high-quality questions. We begin by motivating our choice of ontology and demonstrating its coverage properties (Section 3.3). We then describe our crowdsourcing methodology for soliciting questions and question-article relevance judgments (Section 3.4), and outline our model for determining the relevance of these questions to new text (Section 3.5). After this we describe the two datasets that we construct for the evaluation of our approach and present quantitative results (Section 3.6) as well as examples of our output and an error analysis (Section 3.7) before concluding (Section 3.8).

## 3.2 Related Work

We consider three aspects of past research in automatic question generation: work that focuses on the grammaticality of natural language question generation, work that focuses on the semantic quality of generated questions, i.e., the “what to ask about” rather than “how to ask it,” and finally work that builds a semantic representation of text in order to generate higher-level questions. Approaches focusing on the grammaticality of question generation date back to the AUTOQUEST system [202], which examined the generation of Wh-questions from single sentences. Later systems addressing the same goal include methods that use transformation rules [129], template-based generation [34, 41] and overgenerate-and-rank methods [75]. Another approach has been to create fill-in-the-blank questions from single sentences to ensure grammaticality [2, 11]. More relevant to our direction is work on the semantic aspect of question generation, which has become a more active research area in the past several years. Several authors [118, 109] generate questions according to the semantic role patterns extracted from the source sentence. Becker et al. [11] also leverage semantic role labeling within a sentence in a supervised setting. We hope to continue in this direction of semantic focus, but extend the capabilities of question generation to include open-ended questions that go far beyond the scope of a single sentence. Other work has taken on the challenge of deeper questions by attempting to build a semantic representation of arbitrary text. This has included work using concept maps over keywords [135] and minimal recursion semantics [205] to reason over concepts in the text. While the work of [135] is impressive in its possibilities, the range of the types of questions that can be generated is restricted by a relatively specific set of relations (e.g., Is-A, Part-Of) captured in the ontology

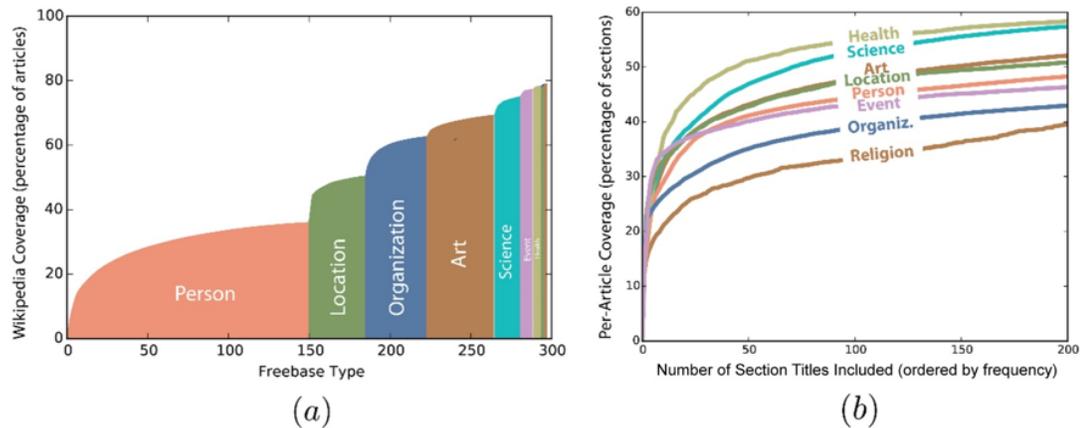


Figure 3.2: Coverage properties of our category-section representation: (a) fraction of Wikipedia articles covered by the top  $j$  most common Freebase types, grouped by our eight higher-level categories. (b) Average fraction of sections covered per document if only the top  $k$  most frequent sections are used; each line represents one of our eight categories.

of the domain (biology textbook). Mannem et al. [117] observe as we have that “capturing the exact true meaning of a paragraph is beyond the reach of current NLP systems;” thus, in their system for Shared Task A (for paragraph-level questions [161]) they make use of predicate argument structures along with semantic role labeling. However, the generation of these questions is restricted to the first sentence of the paragraph. Though motivated by the same noble impulses of these authors to achieve higher-level questions, our hope is that we can bypass the challenges and constraints of semantic parsing and generate deep questions via a more holistic approach.

### 3.3 An ontology of categories and sections

The key insight of our approach is that we can leverage an easily interpretable (for crowd workers), low-dimensional ontology for text segments in order to crowdsource a set of high-level, reusable templates that generalize well to many documents. The choice of this representation must strike a balance between domain coverage and the crowdsourcing effort required to obtain that coverage. Inasmuch as Wikipedia is deemed to have broad coverage of human knowledge, we can estimate domain coverage by measuring what fraction of that corpus is covered by the proposed representation. In this chapter, we have developed a category-section ontology using annotations from Freebase and Wikipedia (English), and now describe its structure and coverage in detail. For the high-level categories, we make use of the Freebase “notable type” for each Wikipedia article. In contrast to the noisy default Wikipedia categories, the Freebase “notable types” provide a clean high-level encapsulation of the topic or entity discussed in a Wikipedia article. As we wish to maximize coverage, we compute the histogram by type and take the 300 most common ones across Wikipedia. We further merge these into eight broad categories to reduce crowdsourcing effort: Person, Location, Event, Organization, Art, Science, Health, and Religion. These eight categories cover 78% of Wikipedia articles (see Figure 3.2a); the mapping between Freebase types and our categories will be made available as part of our corpus. To achieve greater specificity of questions within the articles, we make use of Wikipedia sections, which offer a high-level segmentation of the content. The Cartesian product of our categories from above and the most common Wikipedia section titles (per category) then yield an interpretable, low-dimensional representation of the article. For instance, the set of category-section

<b>Person</b>	<b>Location</b>	<b>Organization</b>	<b>Art</b>
Early life	History	History	Plot
Career	Geography	Geography	Reception
Pers. life	Economy	Academics	History
Biography	Demographics	Demographics	Production
<b>Science</b>	<b>Event</b>	<b>Health</b>	<b>Religion</b>
Description	Background	Treatment	Etymology
Taxonomy	Aftermath	Diagnosis	Iconography
History	Battle	Causes	Worship
Distribution	Prelude	History	Mythology

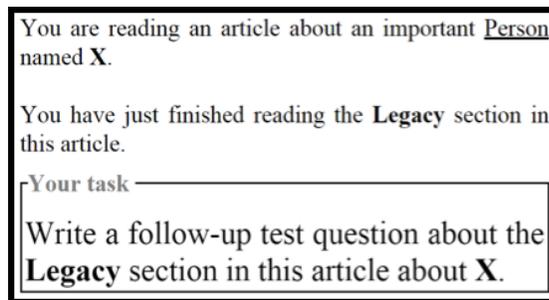
Table 3.1: Most frequent section titles by category.

pairs for an article about Albert Einstein contains (Person, Early life), (Person, Awards), and (Person, Political\_views) as well as several others. For each category, the section titles that occur most frequently represent central themes in articles belonging to that category. We therefore hypothesize that question templates authored for such high-coverage titles are likely to generalize to a large number of articles in that category. Table 3.1 below shows the four most frequent sections for each of our eight categories. As the crowdsourcing effort is directly proportional to the size of the ontology, our goal is to select the smallest set of pairs that will provide sufficient coverage. As with categories, the cut-off for the number of sections used for each category is guided by the trade-off between coverage and crowdsourcing costs. Figure 3.2b plots the average fraction of an article covered by the top k sections from each category. We found that the top 50 sections cover 30% to 55% of the sections of an individual article (on average) across our categories. This implies that by only crowdsourcing question templates for those 50 sections per category, we would be able to ask questions about a third to a half of the sections of any article. Of course, if we were to limit ourselves to only segments with these labels at runtime, we would completely miss many articles as well as texts outside of Wikipedia. To extend our reach, we also develop the means for category and section inference from raw text in Section 3.5 below, for

cases in which ontological labels are either not available or are not contained within our limited set.

### 3.4 Crowdsourcing methodology

We designed a two-stage crowdsourcing pipeline to (i) collect templates targeted to a set of category-section pairs and (ii) obtain binary relevance judgments for the generated templates in relation to a set of article segments (for Wikipedia, these are simply sections) that match in category-section labels. We recruit Mechanical Turk workers for both stages of the pipeline, filtering for workers from the United States due to native English proficiency. A total of 307 unique workers participated in the two tasks combined (78 and 229 workers for the generation and ratings tasks respectively).



The image shows a text box with a black border containing the following text:  
You are reading an article about an important Person named **X**.  
You have just finished reading the **Legacy** section in this article.  
Your task \_\_\_\_\_  
Write a follow-up test question about the **Legacy** section in this article about **X**.

Figure 3.3: Prompt for the generation task for the category-section pair (Person, Legacy).

### 3.4.1 Question generation task

Following the coverage analysis above, we select the 50 most frequent sections for the top two categories, Person and Location, yielding 100 category-section pairs. As these two categories cover nearly 50% of all articles on Wikipedia, we believe that they suffice in demonstrating the effectiveness of the proposed methodology. For each category-section pair, we instructed 10 (median) workers to generate a question regarding a hypothetical entity belonging to the target with the prompt in Figure 3.3. Additional instructions and an interactive tutorial were pre-administered, guiding the workers to formulate appropriately deep questions, i.e., questions that are likely to generalize to many articles, while avoiding factoid questions like When was X born? In total, 995 question templates were added to our question database using this methodology (only 0.5% of all generated questions were exact repeats of existing questions). We confirm in Section 4.2 that workers were able to formulate deep, interesting and relevant questions whose answers spanned more than a single sentence and that generalized to many articles using this prompt. In earlier pilots, we tried an alternative prompt which also presented the text of a specific article segment. In Figure 3.4, we show the average scope and relevance of questions generated by workers under both prompt conditions. As the figure demonstrates, the alternative prompt showing specific article text resulted in questions that generalized less well (workers questions were found to be relevant to fewer articles), likely because the details in the text distracted the workers from thinking broadly about the domain. These questions also had a smaller scope on average, i.e., answers to these questions were contained in shorter spans in the text. The differences in scope and relevance between the two prompt designs were both significant ( $p$ -values: 0.006 and  $4.5e-11$  respectively, via two-sided Welch's  $t$ -tests).

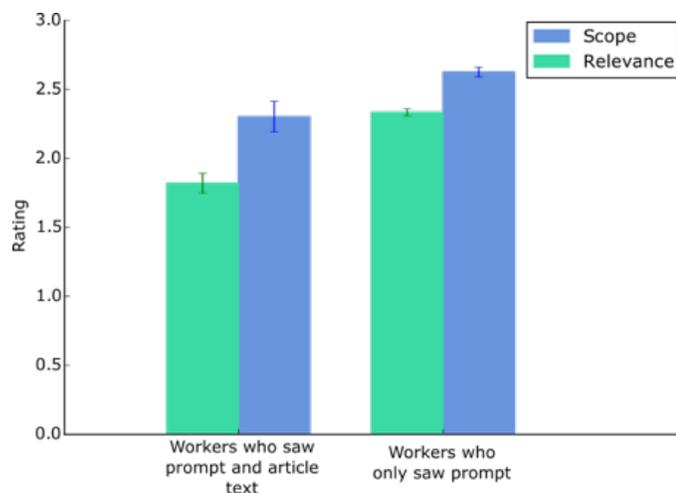


Figure 3.4: Average relevance and scope of worker-generated questions versus how the workers were prompted.

### 3.4.2 Question relevance rating task

For our 100 category-section pairs, 4 (median) article segments within reasonable length for a Mechanical Turk task (200-1000 tokens) were drawn at random from the Wikipedia corpus; this resulted in a set of 513 article segments. Each worker was then presented with one of these segments alongside at most 10 questions from the question template database matching in category-section; templates were converted into questions by filling in the article-specific entity extracted from the title. Workers were requested to rate each question along three dimensions: relevance, quality, and scope, as detailed below. Quality and scope ratings were only requested when the worker determined the question to be relevant.

- **Relevance:** 1 (not relevant) 4 (relevant) Does the article answer the question?
- **Quality:** 1 (poor) 4 (excellent) Is this question well-written?

- **Scope:** 1 (single-sentence) 4 (multi-sentence/paragraph) How long is the answer to this question?

A median of 3 raters provided an independent judgment for each question-article pair. The mean relevance, quality and scope ratings across the 995 questions were 2.3 (sd=0.83), 3.5 (sd=.65) and 2.6 (sd=1.0) respectively. Note that the sample sizes for scope and quality were smaller, 774 and 778 respectively, as quality/scope judgments were not gathered for questions deemed irrelevant. We note that 80% of the relevant crowd-sourced questions had a median scope rating larger than 1 sentence, and 23% had a median scope rating of 4, defined as the answer to this question can be found in many sentences and paragraphs, corresponding to the maximum attainable scope rating. Note that while in this chapter, we have only used the scope judgments to report summary statistics about the generated questions, in future work these ratings could be used to build a scope classifier to filter out questions targeting short spans of text. As described in Section 3.5.2, the relevance judgments are converted to binary relevance ratings for training the relevance classifier (we consider relevance ratings 1, 2 as not relevant and 3, 4 as relevant). In terms of agreement between raters for these binary relevance labels, we obtained a Fleiss Kappa of 0.33, indicating fair agreement.

### 3.5 Model

There are two key models to our system: the first is for category and section inference of a novel article segment, which allows us to infer the keys to our question database when explicit labels are not available. The second is for question relevance prediction, which lets us decide which question templates

from the databases store for that category-section actually apply to the text at hand.

### 3.5.1 Category/section inference

Both category and section inference were cast as standard text-classification problems. Category inference is performed on the whole article, while section inference is performed on the individual article segments (i.e., sections). We trained individual logistic regression classifiers for the eight categories and the 50 top section types for each one (a total of 400) using the default L2 regularization parameter in LIBLINEAR [50]. For section inference, a total of 736,947 article segments were sampled from Wikipedia (June 2014 snapshot), each belonging to one of the 400 section types and within the same length bounds from Section 3.4.2 (200-1000 tokens). For category inference, we sampled a total of 86,348 articles with at least 10 sentences and belonging to one of our eight categories. In both cases, a binary dataset was constructed for a one-against-all evaluation, where the negative instances were sampled randomly from the negative categories or sections (there was an average 17% and 32% positive skew in the section and category datasets, respectively). Basic tf-idf features (using a vocabulary of 200,000 after eliminating stopwords) were used in both text classification tasks. Applying the category/section inference to held-out portions of the dataset (30% for each category/section) resulted in balanced accuracies of 83%/95% respectively, which gave us confidence in the inference. Keep in mind that this is not a strict bound on our question generation performance, since the inferred category/section, while not matching the label perfectly, could still be sufficiently close to produce relevant questions (for instance, we could

misrecognize Childhood as Early Life). We explore the ramifications of this in our end-to-end experiments in Section 3.6.

### 3.5.2 Relevance classification

We also cast the problem of question/article relevance prediction as one of binary classification, where we map a question-article pair to a relevance score; as such our features had to combine aspects of both the question and the article. Our core approach was to use a vector of the component-wise Euclidean distances between individual features of the question and article segment, i.e., the  $i$ th feature vector component is given by  $\sqrt{(q_i - a_i)^2}$ , where  $q_i$  and  $a_i$  are the components of the question and article feature vectors. For the feature representation, we utilized a concatenation of continuous embedding features: 300 features from a Word2Vec embedding [124] and 200,000 tfidf features (as with category/section classification above). As question templates are typically short, though, we found that this representation alone performed poorly. As a result, we augmented the vector by concatenating additional distance features between the target article segment and one specific instance of an entire article for which the question applied. This augmenting article was selected at random from all those for which the template was judged to be relevant. The resulting feature vector was thus doubled in length, where the first distances were between the question template and the target segment, and the next were between the augmenting article and the target segment. Note that the augmenting article segments were removed from the training/test sets. To train this classifier, we assumed that we would be able to acquire at least positive relevance labels for each question template, i.e., article segments judged to be relevant to each template for inclusion in the training set. We explore the

effect of increasing values of  $n$ , from 0 (where no relevance labels are available) to 3 (referred to as conditions T0..T3 in Figure 3.5). We then trained and evaluated the relevance classifier, a single logistic regression model using LIBLINEAR with default L2 regularization, using 10-fold cross-validation on dataset I (see Section 3.6). Figure 3.5 depicts a series of ROC curves summarizing the performance of our template relevance classifier on unseen article segments. As expected, we see increasing performance with increasing  $n$ . However, the benefit drops off after 3 instances (i.e., T4 is only marginally better than T3). While the character of the curves is modest, keep in mind we are already filtering questions by retrieving them from the database for the inferred category-section (which by itself gives us a precision of .74 see green bars in Figure 3.6); this ROC represents the lift achieved by further filtering the questions with our relevance classifier, resulting in far higher precision (.85 to .95 see blue bars in Figure 3.6).

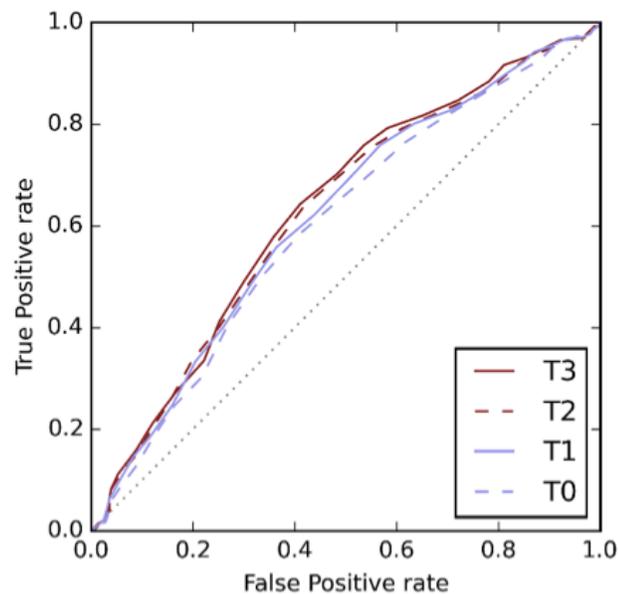


Figure 3.5: ROC curves for the task of question-to-article relevance prediction.  $T_n$  means that  $n$  positively labeled article segments were available for each question template during training.

## 3.6 Experiments and results

In this section, we describe the datasets used for training the relevance classifier in Section 3.5.2 (dataset I) as well as for end-to-end performance on unlabeled text segments (dataset II). We then evaluate the performance on this second dataset under three settings: first, when the category and section are known, second, when those labels are unavailable, and third, when neither the labels nor the relevance classifier are available.

### 3.6.1 Dataset I: for relevance classification

The first dataset (dataset I) was intended for training and evaluating the relevance classifier, and for this we assumed the category and section labels were known. As such, judgments were collected only for questions templates authored for a given articles actual category and section labels. After filtering out annotations from unreliable workers (based on their pre-test results) as well as those with inter-annotator agreement below 60%, we were left with a set of 995 rated questions, spanning across two categories (Person and Location) and 50 sections per category (100 category-section pairs total). This corresponded to a total of 4439 relevance tuples (label, question, article) where label is a binary relevance rating aggregated via majority vote across multiple raters. The relevance labels were skewed towards the positive (relevant) class with 63% relevant instances. This is of course a mostly unrealistic data setting for applications of question generation (known category and section labels), but greatly useful in developing and evaluating the relevance classifier; we thus used this dataset only for that purpose (see Section 3.5.2 and Figure 3.5).

### 3.6.2 Dataset II: for End-to-end evaluation

For an end-to-end evaluation we need to examine situations where the category and section labels are not available and we must rely on inference instead. As this is the more typical use case for our method, it is critical to understand how the performance will be affected. For dataset II, then, we first sampled articles from the Wikipedia corpus at random (satisfying the constraints described in Section 3.3) and then performed category and section inference on the article segments. The category  $c$  with the highest posterior probability was chosen as the inferred category, while all section types with a posterior probability greater than 0.6 were considered as sources for templates. Only articles whose inferred category was Person or Location were considered, but given the noise in inference there was no guarantee that the true labels were of these categories. We continued this process until we retrieved a total of 12 articles. For each article segment in these 12, we drew a random subset of at most 20 question templates from our database matching the inferred category and section(s), then ordered them by their estimated relevance for presentation to judges. We then solicited an additional 62 Mechanical Turk workers to a rating task set up according to the same protocol as for dataset I. After aggregation and filtering in the same way, the second dataset contained a total 256 (label, question, article) relevance tuples, skewed towards the positive class with 72% relevant instances.

### 3.6.3 Information Retrieval-based evaluation

As our end-to-end task is framed as the retrieval of a set of relevant questions for a given article segment, we can measure performance in terms of an information

retrieval-based metric. Consider a user who supplies an article segment (the query in IR terms) for which she wants to generate a quiz: the system then presents a ranked list of retrieved questions, ordered according to their estimated relevance to the article. As she makes her way down this ranked list of questions, adding a question at a time to the quiz (set  $Q$ ), the behavior of the precision and recall (with respect to relevance to the article segment) of the questions in  $Q$ , summarizes the performance of the retrieval system (i.e., the Precision-Recall (PR) curve [170]). We summarize the performance of our system by averaging the individual article segments PR curves (linearly interpolated) from dataset II, and present the average precision over bins of recall values in Figure 3.6. We consider the following experimental conditions:

- **Known category/section, using relevance classifier (red):** This is the case in which the actual category and section labels of the query article are known, and only the questions that match exactly in category and section are considered for relevance classification (i.e., added to  $Q$  if found relevant by the classifier). Recall is computed with respect to the total number of relevant questions in dataset II, including those corresponding to sections different from the section label of the article.
- **Inferred category/section, using relevance classifier (blue):** This is the expected use case, where the category/section labels are not known. Questions matching in category and section(s) to the inferred category and section of each article are considered and ranked in  $Q$  by their score from the relevance classifier. Recall is computed with respect to the total number of relevant questions in dataset II.
- **Inferred category/section, ignoring relevance classifier (green):** This is a baseline where we only use category/section inference and then retrieve

questions from the database without filtering: all questions that match in inferred category and section(s) of the article are added to Q in a random ranking order, without performing relevance classification.

As we examine Figure 3.6, it is important to point out a subtlety in our choice to calculate recall of the known category/section condition (red bars) with respect to the set of all relevant questions, including those that are matched to sections different from the original (labeled) sections. While this condition by construction does not have access to questions of any other section, the resulting limitation in recall underlines the importance of performing section inference: without inference, we achieve a recall of no greater than 0.4. As we had hypothesized, while the labels of the sections play an instrumental role in instructing the crowd to generate relevant questions, the resulting questions often tend to be relevant to content found under different but semantically related sections as well. Leveraging the available questions of these related sections (by performing inference) boosts recall at the expense of only a small degree of precision (blue bars). If we forgo relevance classification entirely, we get a constant precision of 0.74 (green bars) as mentioned in Section 3.5.2; it is clear that the relevance classifier results in a significant advantage. While there is a slight drop in precision when using inference, this is at least partly due to the constraints that were imposed during data-collection and relevance classifier training, i.e., all pairs of articles and questions belonged to the same category and section. While this constraint made the crowdsourcing methodology proposed in this chapter tractable, it also prevented the inclusion of training examples for sections that could potentially be inferred at test time. One possible approach to remedy this would be sample from article segments that are similar in text (in terms of our distance metric) as opposed to only segments exactly matching in

category and section.

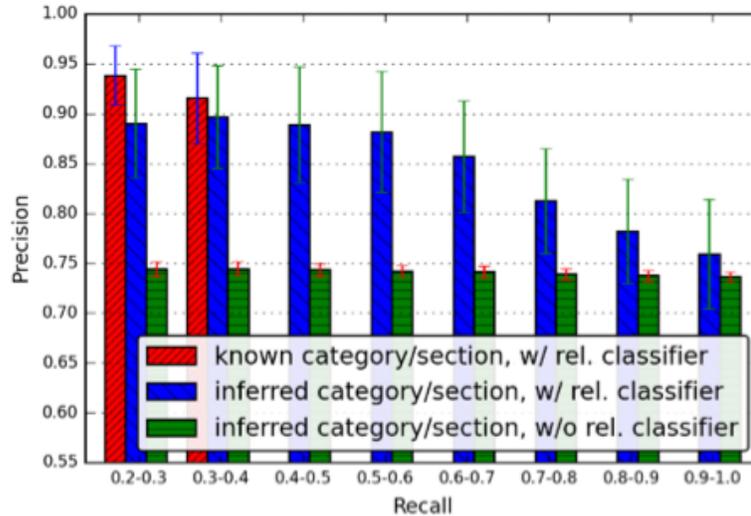


Figure 3.6: Precision-recall results for the end-to-end experiment, grouped in bins of recall ranges.

### 3.7 Examples and error analysis

In Table 3.2 we show a set of sample retrieved questions and the corresponding correctness of the relevance classifiers decision with respect to the judgment labels; examining the errors yields some interesting insights. Consider the false positive example shown in row 8, where the category correctly inferred as Location, but section title was inferred as Transportation instead of Services. This mismatch resulted in the following template authored for (Location, Transportation) being retrieved: “What geographic factors influence the preferred transport methods in <entity>?” To the relevance classifier, this particular template (containing the word transport) appears to be relevant on the surface level to the text of an article segment about schedules (Services) at a railway station. However, as this template never appeared to judges in the context of a Services segment a sec-

True section	Inferred section	Result	Generated Question
Honours	Later Life	TP	What accomplishments characterized the later career of Colin Cowdrey?
Acting Career	Television	TP	How did Corbin Bernsteins television career evolve over time?
Route Description	Geography	TP	What are some unique geographic features of Puerto Rico Highway 10?
Athletics	Athletics	TN	How much significance do people of DeMartha Catholic High School place on athletics?
Route Description	Geography	TN	How does the geography of Puerto Rico Highway 10 impact its resources?
Work	Reception	FN	What type of reaction did Thornton Dial receive?
Acting Career	Later Career	FP	What were the most important events in the later career of Corbin Berstein?
Services	Transportation	FP	What geographic factors influence the preferred transport methods in Weymouth Railway Station?
Later Career	Legacy	FP	How has Freddy Mitchells legacy shaped current events?

Table 3.2: Examples of retrieved questions. TP, TN, FP, FN stand for true/false positive/negative with respect to the relevance classification.

tion that differs considerably in theme from the inferred section (Transportation) the relevance classifier unsurprisingly makes the wrong call.

In considering additional sources of relevance classification errors, recall that we employ a single relevant article segment for the purpose of augmenting a templates feature representation. In the case of the false negative example (row 6 in Table 3.2), the sensitivity of the classifier to the particular augmenting article used is apparent. Upon inspecting the target article segment (article: Thornton Dial, section: Work), and the augmenting article segment (article: Syed Masood,

section: Reception), its clear that the inferred section Reception is a reasonable title for the Work section of the article on Thornton Dial, making the question What type of reaction did Thornton Dial receive? a relevant question to the target article (as reflected in the human judgment). However, although both segments generally talk about reception, the language across the two segments is distinct: the critical reception of Thornton Dial the visual artist is described in a different way from the reception of Syed Masood the actor, resulting in little overlap in surface text, and as a result the relevance classifier falsely rejects the question. Reasonable substitutions for inferred sections can also lead to false positives, as in row 9, for the article Freddy Mitchell. In this case, while Legacy (the inferred section) is a believable substitute for the true label of Later Career, in this case the article segment did not discuss his legacy. However, there was a good match between the augmenting article for this template and the section. We hypothesize that in both this and the previous examples a broader sample of augmenting article segments for each category/section is likely to be effective at mitigating these types of errors.

### **3.8 Conclusion**

We have presented an approach for generating relevant, deep questions that are broad in scope and apply to a wide range of documents, all without constructing a detailed semantic representation of the text. Our three primary contributions are (i) our insight that a low-dimensional ontological document representation can be used as an intermediary for retrieving and generalizing high-level question templates to new documents, (ii) an efficient crowdsourcing scheme for soliciting such templates and relevance judgments (of templates to article) from

the crowd in order to train a relevance classification model, and (iii) using category/section inference and relevance prediction to retrieve and rank relevant deep questions for new text segments. Note that the approach and workflow presented here constitute a general framework that could potentially be useful in other language generation applications. For example, a similar setup could be used for high-level summarization, where question templates would be replaced with “summary snippets.”

## CHAPTER 4

### OPTIMAL ASSESSMENT WITH MULTIPLE CHOICE QUESTIONS

#### 4.1 Introduction

In this chapter, we continue the focus on the task of question generation, but shift focus to multiple choice questions (MCQs). Multiple choice questions are the most common form of assessment in Massive open online courses (MOOCs) today [46], primarily due to their inherent scalability in grading. The design of a good set of options in multiple-choice questions (MCQs), however, is notoriously difficult [157]. Incorrect options, also known as distractors, should ideally be picked from a representative set of misconceptions that students commonly share. But even if this set is representative, the question might still fail to distinguish between students who were “close” to the correct answer, and those who were clueless. In the adaptive testing literature [112, 195], the questions themselves are selected to be at a level that is appropriate for the student, such that their responses result in the most accurate estimate of their knowledge. In this chapter, we pursue the same goal, but at the level of designing a single question, i.e., to select a set of options to present as potential answers. This problem is not a straightforward extension of the classic adaptive testing problem for two reasons: (i) from an application perspective, only recently with the advent of web-scale learning platforms we are able to leverage the massive number of student submissions and answer click-through logs to generate rich, adaptive, and data-driven questions that exploit actual student misconceptions; (ii) from a technical level, selecting choices is inherently a batch optimization problem, i.e.,

---

<sup>0</sup>This chapter has been adapted from the paper [99]

all options must be considered jointly during optimization; this is in stark contrast to question selection, which typically assumes independence between questions and finds the optimal set in a greedy fashion (though test bank optimization is an exception, see Chapter 7 in [53]) . The main contributions of this chapter are summarized as follows:

- We propose an objective function for selecting an optimal set of choices in a discrete choice model, given the estimated user ability, and we investigate the solutions across different regimes of student ability.
- We propose an algorithm for finding a globally-optimal option choice set.
- We collect a dataset used in our experiments: A “U.S. States Quiz” dataset, where users were given an MCQ quiz on their knowledge of U.S. states.
- We propose a new paradigm of data-driven test design by leveraging data from technical online forums, and showcase the applicability of this model to the task of MCQ design from StackExchange posts.

## 4.2 Related Work

### 4.2.1 Education

In the education literature, multiple choice testing has received significant attention, studying a broad range of aspects of MCQ design, e.g., to ensure validity (i.e., does the question measure learning outcomes?) [68, 67], to decide on the optimal number of choices [157, 69], and to design good distractors [70, 67]. In an empirical study [68], Haladyna and Downing concluded that the key in

multiple choice item design was “not the number of distractors but the quality of distractors.” They find that, almost unanimously, high-quality distractors are considered to be those that represent common student misconceptions [69]. Thissen et al. [186] developed a graphical analysis method of distractors based on the response statistics in the context of a nominal item response model, with the goal of facilitating a posteriori analysis of multiple choice items. Computational methods have been proposed for the task of multiple choice item design (i.e., designing a question and its choices), but are restricted to specific domains, such as vocabulary [26], grammar testing [12], or topic-specific comprehension [128]. For all these methods, however, distractors are generated automatically based on the structure of the problem domain. We are unaware of prior results that optimize for a distractor choice set based on the data of past student submissions.

## **4.2.2 Active Learning and Adaptive Testing**

The field of adaptive testing borrows techniques from the areas of active learning and optimal experiment design. Adaptive testing is classically posed as a task of item set optimization (classically in an online setting, see [112, 195]), where the optimization objective is related to the estimator efficiency, typically of student ability (see Chapter 7 of [53] for an overview). More recently, methods based on the principle of estimator efficiency have been applied to the task of test-set reduction [192] in the context of a multidimensional extension of the Rasch model (SPARFA) [194]. We can view choice-set optimization as a non-trivial generalization of optimal test-design that was traditionally explored in the setting of item-set optimization only. We argue that this extension will become particularly relevant in rapidly growing, data-intensive educational settings, where a subset

of real student submissions can be efficiently selected into an optimal distractor set for a student of a specific ability level.

### **4.3 Optimization with discrete choice models**

In marketing and operations research there has been a long history of interest in the problem of experiment design with discrete choice models, motivated by problems of learning about users' preferences towards product attributes (e.g., to understand what features a user favors in a credit card). The problem is typically formulated as finding an experiment design (i.e., a full factorial design or a subset) that consists of a combination of choices sets and attribute levels (e.g., features describing a credit card, such as interest rate, rewards, annual fee) to be administered to each user [30, 89, 164, 80]. An example would be a taste test, where the same user is administered a sequence of the same food item but with various levels of its ingredients (e.g., salt, sugar, cinnamon). A number of heuristic search algorithms for finding optimal designs have been proposed [89, 187, 125, 214, 37] in literature. However, these methods are not directly relevant to the task proposed in this chapter where (i) attribute levels cannot be independently varied, (ii) the same user may not be exposed to an experiment consisting of the same items with different attribute levels.

### **4.4 Model**

In formulating our model, we require it to exhibit the following three properties:

**Property 1** The model specifies a probability of a student choosing a particular option as a function of that student’s *ability* and that option’s *correctness*, such that students of greater ability are more likely to pick the most correct option (we will discuss this aspect in detail below).

**Property 2** A “perfect” student (with the highest attainable *ability*) chooses the correct option with probability 1.

**Property 3** A student with the lowest attainable *ability* makes their choice uniformly at random.

For simplicity, we require that there is exactly one correct option, leaving the remaining options as distractors that lie on a continuum of *apparent correctness*, i.e., options that vary in how difficult they are to discern from the correct answer (and such that a more able student is more likely to discern the correct option).

A multinomial logit model with a partial order constraint on the *apparent correctness* of each choice  $\beta_j$  and a non-negativity constraint on the student’s *ability*  $\theta_i$ , exhibits all of the properties above. Specifically, we use the following statistical model:

$$P(i \text{ picks option } j \mid \theta_i, \{\beta_j\}_{j \in C}) = \frac{\exp(\theta_i \beta_j)}{\sum_{j' \in C} \exp(\theta_i \beta_{j'})}, \quad (4.1)$$

where  $j$  is the option index,  $\beta_{j^*} > \beta_j, \forall j \in C \setminus j^*$ , and  $j^*$  is the correct option. Furthermore, we assume  $\theta_i \geq 0, \forall i$ , where  $\theta_i$  is the ability of student  $i$ , and  $\{\beta_j\}_{j \in C}$  is the set of option parameters presented to the student, encoding the *apparent correctness* of each option. Without an explicit partial order constraint on the choices and a non-negativity constraint on the students, the model would capture the relative preference of subjects towards choices. In psychometrics this model is known as the nominal response model [53] and is also related to

the more general multidimensional unfolding models [38, 168] often used to investigate the relationship between subjects and preferences. In our setting, the non-negativity constraint on the ability  $\theta_i$ , combined with the partial order constraints on the option parameters  $\{\beta_j\}$  are critical to obtaining the desired interpretation of the ability parameter  $\theta_i$ , namely as capturing the *ability* of the student (larger values indicate greater ability). One can easily verify that Property 2 and Property 3 are both satisfied by considering the limiting behavior of (4.1) when  $\theta_i = 0$  and  $\theta_i = \infty$  respectively. Property 1 is satisfied as a result of  $P(i \text{ picks option } j^* \mid \cdot)$  (i.e., the probability of student  $i$  picking a correct option) being a monotone function of  $\theta_i$ . As a consequence, performing optimal option subset selection under this model and these constraints will result in subsets that are most informative about the students' *abilities*.

It is also important to understand the limitations and additional assumptions underlying this model. The most significant limitation is what is known as the *independence of irrelevant alternatives* (IIA) assumption [153, 114]. The IIA assumption is violated whenever the two options are not inherently different. For example, in the setting of reusing student responses as potential options in a test, this would occur if the two options are either completely identical or are paraphrases of each other. We leave dealing with the problem of IIA to future work.

To place our model in the context of existing work, we compare it with two closely related models: the classical Rasch model [53] and the recent model proposed by Bachrach *et al.* [7].

### 4.4.1 Relationship to the Rasch model

The classical dichotomous Rasch model defines the likelihood of a student answering a question correctly as a function of the question's difficulty and the student's ability, i.e., it is agnostic to the actual choice made by the student in an MCQ setting. The likelihood of student  $i$  with ability  $\theta_i$  getting the question  $j$  with difficulty  $q_j$  correct is given by:

$$P(i \text{ correctly answers } j \mid \theta_i, q_j) = \frac{1}{1 + \exp(-(\theta_i - q_j))}.$$

To gain intuition about how our model encodes question *difficulty*, consider the case of only two options: the correct option with parameter  $\beta_{j^*}$  and the incorrect option with parameter  $\beta_j$ . We can now express the likelihood of the student answering this question correctly using our model as follows:

$$P(i \text{ correctly answers } j \mid \theta_i, \Delta_{j^*-j}) = \frac{1}{1 + \exp(-\theta_i \Delta_{j^*-j})},$$

where  $\Delta_{j^*-j} = \beta_{j^*} - \beta_j$ , which is positive by definition (since  $\beta_{j^*} > \beta_j$ ). By analogy with the Rasch likelihood,  $\Delta_{j^*-j}^{-1}$  captures a similar notion of question *difficulty*: the farther apart are the two options in the parameter space, the "easier" is the resulting question.

When the question contains more than two options, the likelihood of the student answering the question correctly can be expressed as:

$$P(i \text{ right on } j \mid \theta_i, \{\Delta_{j^*-j}\}_j) = \frac{1}{1 + \sum_{j \in Q} \exp(-\theta_i \Delta_{j^*-j})},$$

where an exponential term containing the distance  $\Delta_{j^*-j}$  between the correct option and every remaining option now appears in the denominator. Observe that the probability of getting the question right approaches one only when the correct option parameter (scaled by ability  $\theta_i$ ) is well-separated from every other

option (distractor). An important advantage offered by modeling individual choices is that the model’s estimate of the students’ abilities will not only depend on which questions were answered correctly and incorrectly, but also on the nature of the incorrect answers chosen. Consequently, our model could distinguish between the abilities of two students, even if both of these students answered all questions incorrectly.

#### 4.4.2 Relationship to Bachrach et al.

Recently Bachrach *et al.* [7] extended the dichotomous Rasch to account for the observation of the actual choice, with the goal of inferring the correct answers from choice click-through alone (i.e., in an unsupervised way). The (simplified) generative process of their model is defined as follows:

$$z_{ij} \sim P(i \text{ correctly answers } j \mid \theta_i, q_j)$$

$$P(i \text{ picks option } k \mid z_{ij}, \pi_k) = \begin{cases} \pi_k & \text{if } z_{ij} = 1 \\ 1/K & \text{otherwise,} \end{cases}$$

which can be interpreted as a mixture model of two components: (i) if the student answers the question correctly ( $z_{ij} = 1$ ), the student picks option  $k$  with probability  $\pi_k$  and (ii) if the student answers the question incorrectly ( $z_{ij} = 0$ ), the student picks an option uniformly at random (i.e.,  $\pi_k = 1/K$  where  $K$  is the number of options). The probability of the student answering correctly  $P(z_{ij} = 1 \mid \theta_i, q_j)$  is parametrized by the standard Rasch model described in Section 4.4.1<sup>1</sup>.

---

<sup>1</sup>This is a slight oversimplification of the original model (ignoring question “discriminability” parameter) proposed by Bachrach *et al.*, but captures its key aspects for our purpose.

During learning, the parameters  $\{\pi_j\}_j$  are learned for each choice, in addition to  $q_j$  (question difficulty) and  $\theta_i$  (user ability). Both our model and this model try to accomplish the same thing: model student’s choice as a function of the student and the question. While on the surface the two models may appear different, they are closely related. In the above, the marginal likelihood distribution over choices is a discrete distribution that is a convex combination of a uniform discrete distribution and  $\{\pi_j\}_j$  that varies between the two extremes depending on the ability of the student. In our model, the student’s likelihood distribution over choices is a convex combination of a uniform distribution and a Kronecker delta whose mass of one is at the correct choice. However, if we introduce a prior over the  $\beta_j$  parameters in our model and marginalize  $\beta_j$ , the models would be equivalent with respect to the likelihood function, and the only difference would be in the multiplicative rather than linear parametrization of the likelihood. It is worthwhile to note that our model can be used in place of the model by Bachrach *et al.* in a fully unsupervised setting, i.e., without the partial order constraints on choices, in which case it is equivalent to the traditional discrete choice model.

While both our model and the model by Bachrach *et al.* can be used for the task of estimating student and question parameters in the absence of annotated data (i.e., answer key), the fundamental distinction lies in the capability of our formulation to be used for the task of optimal choice set design—a task that is not feasible with the model by Bachrach *et al.* The underlying reason for this distinction is because our model implicitly couples choice parameters and question difficulty (see Section 4.4.1), allowing us to tune question difficulty to students of varying ability levels by optimizing over choice sets. In contrast, in the Bachrach *et al.* model, the question difficulty and choice parameters are decoupled, making it impossible to derive an objective that relates the expected

informativeness of a question about a student and a set of presented choices.

## 4.5 Optimal Choice Sets

We formulate the problem of optimal choice set design as active learning—query a user (student) with an instance (choice set) such that the expected outcome (student’s answer) maximizes information about the unknown parameters (student ability). Given a question’s complete set of potential answer options  $Q$ , a student with ability  $\theta$  is presented with a subset  $C \subseteq Q$ . We are interested in finding a subset  $C^* \subseteq Q$  that is optimal in some sense for the user with a given ability. Specifically, we are interested in choosing  $C$  that results in the smallest variance of the maximum likelihood estimator of  $\theta$ , which is equivalent to  $C$  with the maximum Fisher information w.r.t.  $\theta$ :

$$C^* = \operatorname{argmax} \mathcal{I}(\theta; C) \quad (4.2)$$

where Fisher information of set  $C$ ,  $\mathcal{I}(\theta; C)$ , is given by

$$\mathcal{I}(\theta; C) = -\mathbb{E} \left[ \frac{\partial^2}{\partial \theta^2} \log f(\theta; C) \Big| \theta \right]. \quad (4.3)$$

Here,  $f(\theta; C)$  is the likelihood function in (4.1). It can be shown that the solution to the above is the following combinatorial optimization problem<sup>2</sup>:

$$\begin{aligned} & \underset{\{x_n\}}{\text{maximize}} && \frac{\sum_i^N \sum_{j>i}^N x_i x_j (\beta_i - \beta_j)^2 \exp(\theta[\beta_i + \beta_j])}{\sum_i^N \sum_j^N x_i x_j \exp(\theta[\beta_i + \beta_j])} \\ & \text{subject to} && x_n \in \{0, 1\}, \forall n \in Q \\ & && \sum_{n=1}^N x_n \leq K, \end{aligned} \quad (4.4)$$

---

<sup>2</sup>derivation omitted due to space limitation

where  $\{x_n\}_{n=1\dots N}$  are indicator variables ( $x_n \in \{0, 1\}$ ) that select choices from  $Q$  to be included in  $C$ ,  $N = |Q|$  (i.e., the total number of potential options) and  $K$  is the maximum permissible size of  $C$  (e.g., four options).

To gain insight about the types of choice sets this problem “prefers,” it is instructive to consider a case with only two choices  $\{\beta_i, \beta_j\}$ . In this case the objective reduces to:

$$\frac{\Delta_{ij}^2}{\exp(-\theta\Delta_{ij}) + \exp(\theta\Delta_{ij}) + 2}. \quad (4.5)$$

where we have defined  $\Delta_{ij} = \beta_i - \beta_j$ . From the above expression, we can see that the Fisher information grows approximately as  $\Delta_{ij}^2$  for small values of  $\Delta_{ij}$ , and decays exponentially for larger values of  $\Delta_{ij}$ , with an optimal  $\Delta_{ij}$  depending on the value of  $\theta$ . This behavior resembles the traditional adaptive setting of maximizing information in a Rasch model, where we explicitly choose a single question of optimal difficulty (as opposed to implicitly tuning the question’s difficulty through its choices as we do here). We can gain additional insight into the relationship between student ability  $\theta$  and the optimal spacing of the two choices  $\Delta_{ij}$ . The maximum of the above expression is a solution to:

$$\frac{\theta\Delta_{ij} + 2}{\theta\Delta_{ij} - 2} = \exp(\theta\Delta_{ij}).$$

Consider the case where the student’s ability  $\theta$  increases. It becomes clear from the above that  $\Delta_{ij}$  has to consequently decrease to maintain equality. Intuitively, we gain most about the student’s ability by showing more difficult distractors to more able students and vice-versa. See Figure 9.2 for a geometric intuition behind the optimization problem.

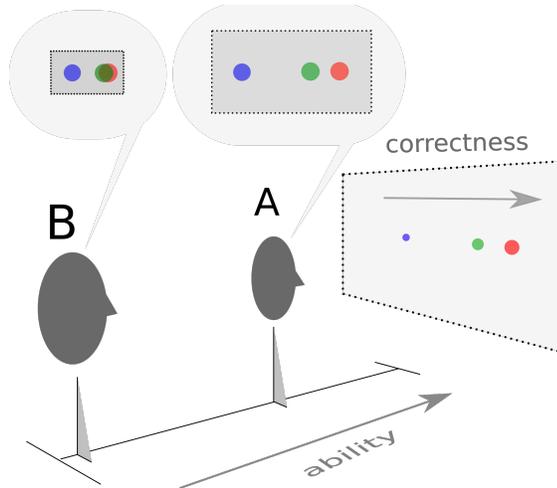


Figure 4.1: Geometric intuition behind the choice of an optimal set of options in a multiple choice test: we imagine the subject’s distance from the wall to be inversely related to their “ability to see.” If asked which colored dot painted on the wall is the right-most dot, subjects closer to the wall would be more likely to answer the question correctly, and subjects farther away would be most likely to guess. The question of optimal choice set design can then be posed as: “where on the wall do we paint the dots, to most efficiently learn about your distance to the wall?”

### 4.5.1 Asymptotically optimal choices

We now investigate the nature of the optimal choice sets. Consider two limiting cases: a student with a large ability ( $\theta_i \rightarrow \infty$ ), and a student with a low ability ( $\theta_i \rightarrow 0$ ).

**Case  $\theta_i \rightarrow \infty$ :** It is straightforward to show that in the limit of “infinite ability,” the information will go to zero. However, the rate at which it goes to zero depends on the choice set, allowing us to gain insight into the kinds of choice sets that will be “preferred” for users with a large ability. The logarithm of the information function will have a linear asymptote, with the slope dominated by the largest exponential in the numerator and the denominator. We can show that

as  $\theta \rightarrow \infty$ , only the two choices with the largest values of  $\beta$  remain relevant (i.e.,  $\{\beta_{max}, \beta_{max-1}\}$ ), with the optimal spacing between them,  $\beta_{max} - \beta_{max-1}$ , given by:

$$\lim_{\theta \rightarrow \infty} \log F(\theta) = 2 \log(\beta_{max} - \beta_{max-1}) + \theta (\beta_{max} + \beta_{max-1} - 2\beta_{max}),$$

where  $\beta_{max}$  indicates the largest  $\beta$  in the set and  $\beta_{max-1}$  as the second largest  $\beta$ . Maximizing with respect to  $\beta_{max-1}$  yields

$$\beta_{max} - \beta_{max-1} = \frac{2}{\theta}.$$

Clearly, the greatest Fisher information for large values of  $\theta$  will be obtained when  $\beta_{max-1} \approx \beta_{max}$ , i.e., when the distance between the two top choices approaches zero.

**Case  $\theta_i \rightarrow 0$ :** In the limiting case of  $\theta \rightarrow 0$ , the objective reduces to:

$$\text{maximize } \frac{1}{K^2} \sum_k^K \sum_{k' > k}^K (\beta_k - \beta_{k'})^2,$$

where  $K$  is the number of options we seek to display to the student and  $k$  indexes over those options. The solution to the above can be obtained by choosing a subset of the choices from  $Q$  with the smallest  $\beta$  (“left-most” or “incorrect” choices) and a subset of choices from  $Q$  with the largest  $\beta$  (“right-most” or “correct” choices) (proof omitted). The intuition behind this solution requires some explanation. It is instructive to consider the optimal solution in the case of only two choices. The optimal “spacing” between the correct choice and the distractor ( $\Delta_{ij}$ ) will lie somewhere between 0 and  $\infty$ , but where exactly depends on our prior belief about the ability of the student ( $\theta$ ). An intuitive interpretation of this solution can be gained by relying on a related notion of *information gain*:

the expected distance (KL-divergence) between the prior and the posterior (after observing the choice) on  $\theta$  (ability). Information gain exhibits the same limiting behavior: when the two choices are infinitely far apart ( $\Delta_{ij} \rightarrow \infty$ ), the student will always pick the correct option regardless of their ability—thus, the posterior will not be updated as a consequence of their choice (hence, no information gain). In the extreme of the two choices spaced very close together, the student will always “flip a coin” between them, again giving away no information about their ability. It is this last scenario that will be fundamental to understanding the optimal choice set (with more than two choices) when  $\theta = 0$ .

Consider now introducing additional choices into the choice set. Appealing to the *information gain* interpretation, we again consider the prior-posterior gain of each potential choice (there are  $K$  of them now). As in the case with only two choices, the prior-posterior gain for each option will be non-zero if the student has a “more than a coin-flip” chance of choosing the better option (i.e., giving the student an opportunity to demonstrate their ability), from which it follows that the remaining options must be sufficiently far apart for a student with  $\theta \approx 0$ . Because under the prior of  $\theta = 0$  each outcome (choice) is equally likely, the expected information gain is a sum of such prior-posterior gains. It follows then that the spacing configuration that maximizes total inter-choice distance will also maximize the expected information gain. It also explains why there should be a large “deadzone” (i.e., no other choices) between the choices separated at the “correct” and the “incorrect” extremes: inserting even a single choice in the middle will result in the student with  $\theta \approx 0$  flipping a coin between the choices at the “correct” extreme and the choice in the middle, neutralizing all of the prior-posterior gain.

## 4.6 Algorithm for finding optimal choice sets

The mathematical programming formulation (4.4) for finding the optimal choice set has binary decision variables and even when these are relaxed to take on real values between 0 and 1, the objective function is a nonlinear function. In this section we describe a practical way of finding an optimal solution, which uses an Integer Linear Programming (ILP) solver as a subprocedure. The idea is to introduce new binary variables that represent the product of two decision variables (which can be enforced using linear constraints), replace the objective function by just the numerator of the original objective function, and add a constraint that bounds the denominator of the original objective function. This problem given in (4.6) is an ILP, for which we invoke the subroutine. The (basic) procedure now is the following: an upper bound on the denominator is given (at first infinity), the best solution is found, given that bound (which can be found using an ILP), then the bound is lowered to slightly below the denominator given by the current solution. This is repeated until all possible denominators are considered. The best overall solution is kept.

$$\begin{aligned}
& \text{maximize } z = \sum_{i,j:i < j} y_{ij}(\beta_i - \beta_j)^2 \exp(\theta[\beta_i + \beta_j]) \\
& \text{subject to } \sum_{i,j} y_{ij} \exp(\theta[\beta_i + \beta_j]) \leq B \\
& \quad y_{ij} \leq x_i, \forall i, j \\
& \quad y_{ij} \leq x_j, \forall i, j \\
& \quad y_{ij} \geq x_i + x_j - 1, \forall i, j \\
& \quad \sum x_i \leq K \\
& \quad x_i \in \{0, 1\}, \forall i \\
& \quad y_{ij} \in \{0, 1\}, \forall i, j
\end{aligned} \tag{4.6}$$

### The Algorithm

1. Set  $\delta = 2 \exp(\min_i \beta_i^2)$ .

Set  $B \leftarrow \infty$ , and solve ILP (4.6). (The current solution is denoted by  $y_{ij}$  and  $z$ .)

2. Let  $B_{\text{eff}} \leftarrow \sum_{i,j} y_{ij} \exp(\theta[\beta_i + \beta_j])$ , let  $r_{\text{best}} \leftarrow z/B_{\text{eff}}$ , and let  $B_{\text{best}} \leftarrow B_{\text{eff}}$ .

3. Repeat while  $B_{\text{eff}} > 0$ :

(a) Set  $B \leftarrow \min\{z/r_{\text{best}}, B_{\text{eff}} - \delta\}$ .

Solve ILP (4.6). (The current solution is denoted by  $y_{ij}$  and  $z$ .)

(b) Let  $B_{\text{eff}} \leftarrow \sum_{i,j} y_{ij} \exp(\theta[\beta_i + \beta_j])$ .

If  $B_{\text{eff}} > 0$  and  $z/B_{\text{eff}} > r_{\text{best}}$  then set  $r_{\text{best}} \leftarrow z/B_{\text{eff}}$  and set  $B_{\text{best}} \leftarrow B_{\text{eff}}$ .

## Proof of Correctness

**Claim 1.** [56] For any  $x_i, x_j \in \{0, 1\}$  we have  $x_i x_j = y_{ij}$  for  $y_{ij} \in \{0, 1\}$ , when the following three constraints are satisfied:  $y_{ij} \leq x_i$ ,  $y_{ij} \leq x_j$  and  $y_{ij} \geq x_i + x_j - 1$ .

*Proof.* If  $x_i = 0$  or  $x_j = 0$ ,  $y_{ij}$  has to equal 0 as well because of the first two constraints. If  $x_i = x_j = 1$  then the third constraint forces  $y_{ij}$  to be 1.  $\square$

**Claim 2.** After every execution of the while loop of the algorithm  $r_{\text{best}}$  is equal to the best objective value of (4.4) with the additional constraint  $\sum_{i,j} y_{ij} \exp(\theta[\beta_i + \beta_j]) \geq B_{\text{eff}}^{\text{after}}$ , where the superscript “after” indicates the values at the end of the loop.

*Proof.* Proof by induction on the number of executions of the while loop. The base case is when the while loop is not executed yet (0 executions of the while loop). At that moment  $z$  is the maximum objective value of (4.6) with  $B = \infty$ . So any solution where  $\sum_{i,j} y_{ij} \exp(\theta[\beta_i + \beta_j]) \geq B_{\text{eff}}$  has objective value that does not exceed  $z$ . Therefore the objective of (4.4) (which is equal to the quotient of objective and the constraint) cannot exceed  $r_{\text{best}}$ , under  $\sum_{i,j} y_{ij} \exp(\theta[\beta_i + \beta_j]) \geq B_{\text{eff}}$ .

Induction step: Suppose the claim is true after  $k - 1$  executions of the while loop. In iteration  $k$ ,  $z$  is the maximum objective value of (4.6) with  $B = \min\{z/r_{\text{best}}^{\text{start}}, B_{\text{eff}}^{\text{start}} - \delta\}$ , where the superscript “start” indicates the values at the start of the loop. By the same argument as above the objective of (4.4) cannot exceed  $z/B_{\text{eff}}^{\text{after}}$ , under  $\sum_{i,j} y_{ij} \exp(\theta[\beta_i + \beta_j]) \geq B_{\text{eff}}^{\text{after}}$  and  $\sum_{i,j} y_{ij} \exp(\theta[\beta_i + \beta_j]) \leq B$ . By the induction hypothesis  $r_{\text{best}}^{\text{start}}$  is equal to the best objective value of (4.4) with the additional constraint  $\sum_{i,j} y_{ij} \exp(\theta[\beta_i + \beta_j]) \geq B_{\text{eff}}^{\text{start}}$ . Because  $r_{\text{best}}^{\text{after}}$  is set to the maximum of these values, we have proved the claim as long as there is no

better solution when  $\sum_{i,j} y_{ij} \exp(\theta[\beta_i + \beta_j]) \geq B$  and  $\sum_{i,j} y_{ij} \exp(\theta[\beta_i + \beta_j]) \leq B_{\text{eff}}^{\text{start}}$ . Note that the choice of  $\delta$  ensures that there is no solution such that  $\sum_{i,j} y_{ij} \exp(\theta[\beta_i + \beta_j]) > B_{\text{eff}}^{\text{start}} - \delta$  and  $\sum_{i,j} y_{ij} \exp(\theta[\beta_i + \beta_j]) < B_{\text{eff}}^{\text{start}}$ . Finally, because  $z$  is an upper bound on the objective value of (4.6) at every execution of the while loop, we know that the denominator can be constrained to be at most  $z/r_{\text{best}}^{\text{before}}$  before we can find an improved solution.  $\square$

**Corollary 1.** *The algorithm above finds a (globally) optimal solution to (4.4).*

### Historical remarks

The first occurrence of replacing products of binary variables by new boolean variables, and using linear constraints to enforce that these have the correct value seems to be Fortet [56]. After this substitution, the problem is a mathematical programming problem where the objective is to maximize the quotient of two linear functions under linear constraints, having  $\{0, 1\}$  variables. This is known as 0-1 hyperbolic programming, introduced in [71] (in fact, the objective in hyperbolic programming may be the sum of quotients of linear functions). An algorithm for the constrained version of this problem is given by Robillard [155], but this needs an additional assumption on the constraint functions, which is not true in this case. Even though many variants of this problem have been studied, to the best of the authors' knowledge, the algorithm proposed here is not yet known.

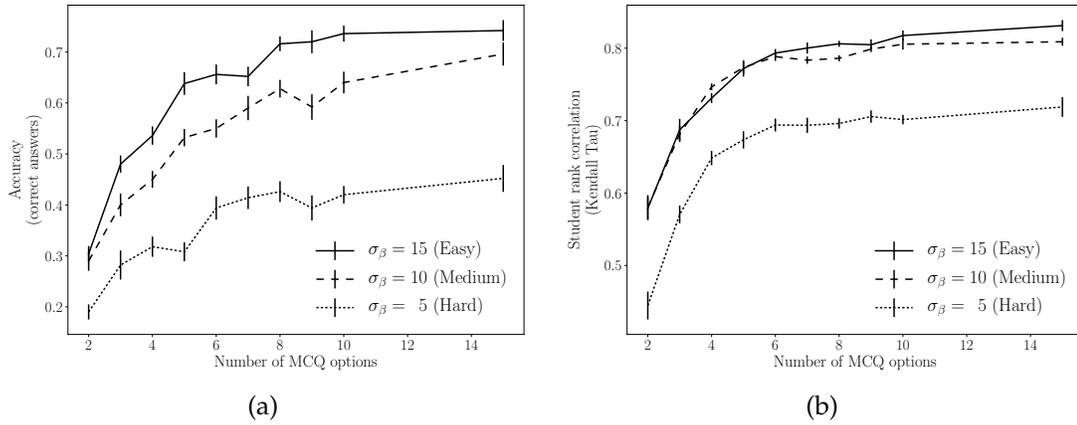


Figure 4.2: Performance (simulation) of the model in (a) predicting the correct answers in a set of questions and (b) ranking students by their ability, as a function of (i) number of choices shown in each question and (ii) variance of the choice parameter distribution (shown as standard deviation  $\sigma_\beta$ ). “Easier” questions correspond to those with a “wider” spread between choice parameters (i.e., higher variance). We can conclude the following on the basis of these results: (i) more choices improves performance in both, predicting correct answers and ranking students, but with diminishing returns, and (ii) showing “easier” questions generally improves performance in correct answer prediction and ranking, however, the model is able to rank well even when the correct answers are more difficult to identify (see Section 4.7). Note that the random baseline for accuracy in (a) is 5% as there are 20 choices for each question in the simulation.

## 4.7 Synthetic Experiments

To gain insight into the behavior of our model, we conduct experiments on synthetic data sampled from the likelihood defined in (4.1). The goal of the synthetic experiments is to (i) validate the correctness of the inference algorithms and (ii) study the effect of the proposed optimal sampling strategy.

### 4.7.1 Parameter learning

The simulation is performed as follows: 100 student ability parameters ( $\theta_i$ ) are sampled from a uniform distribution; 50 questions with 20 options each are generated, where each option parameter  $\beta$  is independently sampled from a zero-mean normal distribution. We evaluate a range of variances for the distribution over choice parameters and study its effect on the quality of the inferred parameters.

We summarize the performance of the inference algorithm via (i) rank correlation of the inferred and ground truth rankings of students and (ii) the accuracy in identifying the correct answers in questions. We use Kendall Tau as a metric of rank correlation. Kendall Tau returns a quantity in the range  $[-1, +1]$ , where  $+1$  indicates perfect correlation (every pair of students in both rankings is in a consistent order),  $-1$  when the rankings are inverted, and  $0$  when the rankings are not correlated. In predicting the correct answer for a question, recall that in our model, the choice with the largest parameter  $\beta$  is interpreted as the correct answer (see Section 4.4). Accuracy in predicting correct answers, therefore, is defined as a fraction of questions where the predicted correct answer matches the ground-truth correct answer.

Figures 4.2(a) and 4.2(b) depict accuracy and rank correlation as a function of the number of choices (i.e., multiple choice options) presented in each question, and as a function of the variance of the distribution over choice parameters  $\beta$ . Recall that the variance of the distribution from which we sample the choice parameters  $\beta$  is inversely related to the difficulty of the resulting question. As we discussed in Section 4.4.1, the question becomes “easy” (i.e., students of lower  $\theta$  will have a high probability of getting it right) when the choice parameters

are “spread out” (which is achieved when the choices are sampled from a high-variance distribution). Both Figure 4.2(a) and Figure 4.2(b) indicate that (i) more choices result in better performance (higher accuracy in identifying correct answers and higher rank correlation between the true and inferred student rankings), and (ii) “easier” questions (i.e., questions whose choice parameters are sampled from a high-variance distribution) generally result in better accuracy and rank correlation.

It is worthwhile to analyze the observation that student rank-correlation (Figure 4.2(b)) remains the same between the “Easy” and “Medium” conditions, while accuracy (Figure 4.2(a)) drops considerably. This can be attributed to the fact that in inferring the ability parameter of a student, the model relies jointly on the parameters of every choice in the set, i.e., not only on whether the chosen option was correct. As a result, while the ordering of the top two choices may be incorrect (resulting in an incorrect prediction of the correct answer), the remaining choices still play an important role in inferring student parameters (and thus in the quality of the ranking).

## 4.7.2 Optimal choice sets

We now evaluate the choice subset selection optimization objective introduced in Section 4.5. We again generate a simulated classroom with 50 students and 50 questions<sup>3</sup>. In contrast to the experiment in Section 4.5, here we perform parameter inference sequentially after each student answers a question, simulating an adaptive testing scenario. For every question, we sample choice sets

---

<sup>3</sup>Student and choice parameters were sampled from uniform distributions with support  $(0, 1)$  and  $(0, 100)$  respectively

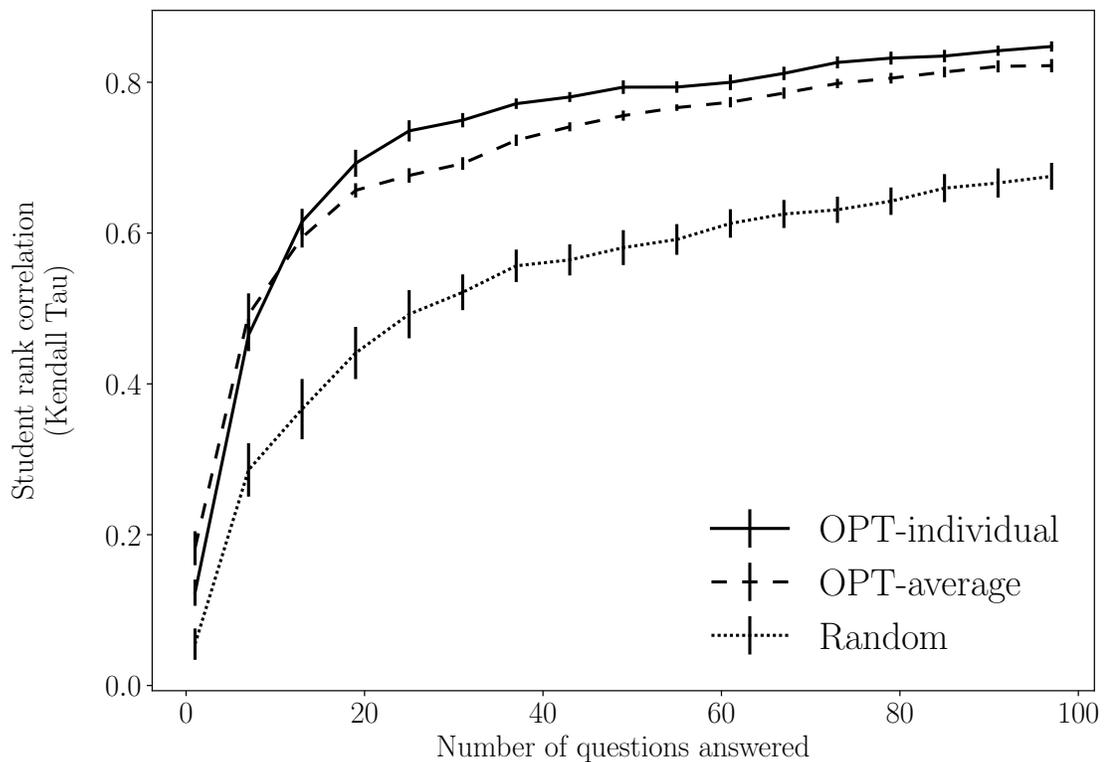


Figure 4.3: Rank correlation between the true and inferred student rankings as a function of the number of questions answered by each simulated student, separated by the choice sampling strategy. Optimizing choice sets according to the proposed objective (**OPT-average** and **OPT-individual**) results in better rank correlation with fewer questions compared to when the choice sets are sampled randomly. Optimizing choice sets according to the individual student abilities (**OPT-individual**) marginally improves performance over optimizing choice sets based on the average student ability (**OPT-average**).

of size 2 according to three different sampling strategies: (i) **random**: choices are drawn uniformly at random, (ii) **OPT-individual**: the optimal choice set is selected for each student according to that student's estimated ability parameter, and (iii) **OPT-average**: the optimal choice set is selected according to the average estimated ability of the student population (i.e., choice sets are identical for each student). Figure 4.3 compares the performance across the three conditions

using the student rank correlation metric introduced in Section 4.7. On the basis of these results, we draw the following conclusions: (i) presenting choice sets optimized using the objective introduced in Section 4.5 with the inferred parameters achieves significantly better rank-correlation and with fewer questions than when the choice sets are sampled randomly; (ii) optimizing choice sets based on the individual student parameters marginally improves performance over optimizing choice-sets to the average ability of the student population. Note, however, that in practice the exact gains will vary depending on the nature of the student and choice parameter distributions.

#### 4.8 User Study: “US States Quiz”

We performed a real-world study to evaluate the importance of data-driven choice set selection in the context of a quiz that asks users to name states of the United States. In this setting, we considered a *question* to be a specific state which the person is required to identify by picking a correct choice out of a set of options (other states). This problem serves as an excellent platform for evaluating our model for two reasons:

1. **Large choice set:** For each state, there are 50 alternatives that can be used as potential options as part of a smaller set of choices.
2. **Ease of evaluation:** The fact that the set of possible answers to each question is finite allows us to use the raw score on a question where all 50 options are presented as the “ground-truth” of the user’s knowledge in this domain. Any other test based on only a subset of the options (and consequently a method used to obtain the options) can be evaluated against this “ground-truth” by

measuring the correlation of the two scores.

3. **Independence of alternatives:** Full independence of alternatives unlikely holds in this setting. Consider, for example, that the user is aware that the state in question is on the west coast and knows which of the options belong to which coast. In this case, adding more options from the east coast should not reduce the probability of that user choosing states on the west coast, which is what would happen in the multinomial logit model. This violation, however, is not as severe as in the case of repeated or paraphrased answers (like in the classic “red bus/blue bus” example).
4. **Large range of “good” and “bad” choices:** Not all distractors in this setting are “created equal”: intuitively we should expect that some states, like those that border the correct state, to be easily mistaken for the correct answer. This provides an opportunity for a data-driven method to excel in finding “good” choice sets for building effective questions.
5. **Common knowledge** Since state knowledge is “common knowledge”, we do not burden the test participants with an additional learning stage, such as a reading comprehension.

#### 4.8.1 Data collection

Mechanical Turk workers residing in the U.S. were solicited to a task titled “How well do you know U.S. states?”, which was briefly described as a quick quiz to test one’s knowledge of the U.S. states, consisting of two stages:

1. **Stage I (fullMCQ):** Workers are presented with a map of the U.S. with a randomly highlighted state and 50 options, one for each state, that they are

required to choose from. This selection is made for every one of the 50 states, presented in random order. Workers are not revealed the correct answer, and are discouraged from looking up the answers externally.

2. **Stage II (subsetMCQ)**: The same workers then repeat the test, but now with only 4 options for each of the 50 states. Options are chosen according to two strategies: **Random** and **Optimal** described in more detail below.

Two experiments were conducted (**Exp1**, **Exp2**) under two different conditions for how the multiple choice options were sampled:

1. (**Exp1**) **Random**: ( $N = 110$ ) During the second stage of the task when only 4 choices are presented (**subsetMCQ**), the choices are selected uniformly at random from the 50 options.
2. (**Exp2**) **Optimal**: ( $N = 67$ ) During the second stage of the task (**subsetMCQ**), the choices are selected according to the optimization objective introduced in Section 4.5. Data collected during the **Random** condition is used to fit the model parameters to be used for optimizing the subsets. The subsets are optimized for the average ability of the users in the **Random** condition (this corresponds to the **OPT-average** strategy introduced in Section 4.7).

## 4.8.2 Evaluation

We propose two strategies for empirically assessing the quality of an MCQ test via two correlation metrics:

1. **Within-subject correlation** The performance of the worker in the first stage of the task (**FullMCQ**) serves as a ground-truth score of that worker's knowledge

of the domain. The correlation of the performance score (fraction of correctly identified states) of the same worker on the same set of questions, but with only a subset of the choices, provides a measure of quality of the presented choice sets.

2. **Between-subject correlation** A good test should also discriminate between workers of different levels of ability. If, for example, student *A* ranks higher than student *B* according to their raw score on the **fullMCQ**, we should expect this ordering to be preserved if we were to instead rank the students based on their performance on the **subsetMCQ** test. We use Kendall Tau—a measure of rank correlation—on students ordered according to their performance on the **fullMCQ** and **subsetMCQ** tests.

## 4.9 Results

### 4.9.1 Within-subject correlation

Figure 4.4 compares the workers' scores according to their performance on the **FullMCQ** and **subsetMCQ** tests, split by condition: **Random** and **Optimal**, where performance is defined as the fraction of states that were named correctly in each test. Both plots indicate that workers with a high score on one test also attain a high score on the other test, which is expected. The critical difference between the two conditions, however, is that of the 40% of the workers that attained a full-score (all correct) on the **subsetMCQ** in the **Random** condition, less than 4% of them attained a full score on the **fullMCQ**.

The **subsetMCQ** test where the choices are generated according to the **Op-**

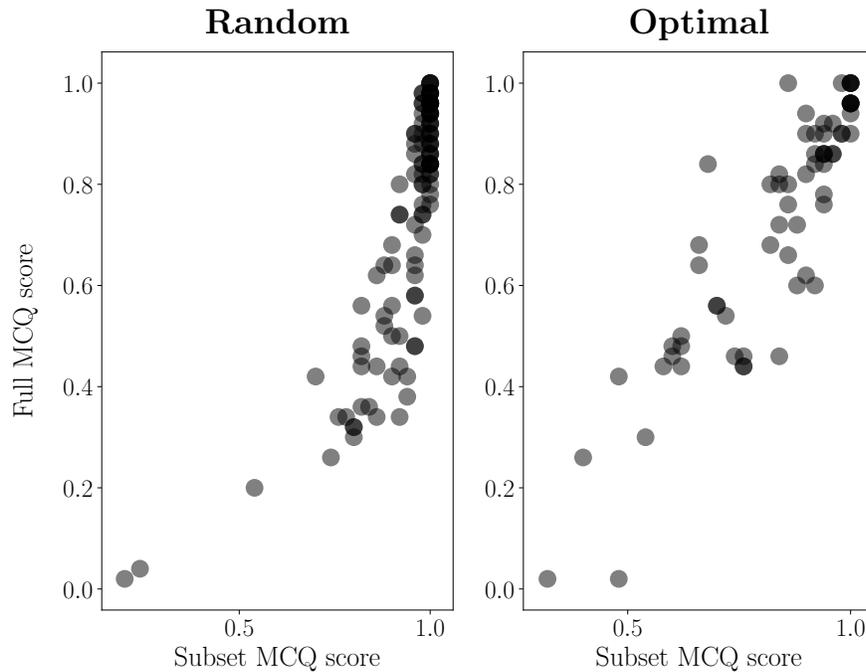


Figure 4.4: Within-subject correlation between raw scores attained on the **subsetMCQ** and **fullMCQ** tests separated by choice set design strategy—choice sets optimized according to the proposed objective yield better within-subject score correlation than choice sets sampled randomly.

**timal** strategy helps remove the full-score bias in the score distribution on the **subsetMCQ** test. Specifically, less than 17% of the workers attain full score on the **subsetMCQ** designed according to the **Optimal** strategy. Additionally, Pearson’s correlation in the **Optimal** condition is 0.89, in contrast to 0.78 in **Random**.

#### 4.9.2 Between-subject correlation

We now focus on the quality of the workers’ ranking using the raw scores obtained on the **subsetMCQ** test between the **Optimal** and **Random** strategies. Our hypothesis is that a test designed to elicit maximum information about

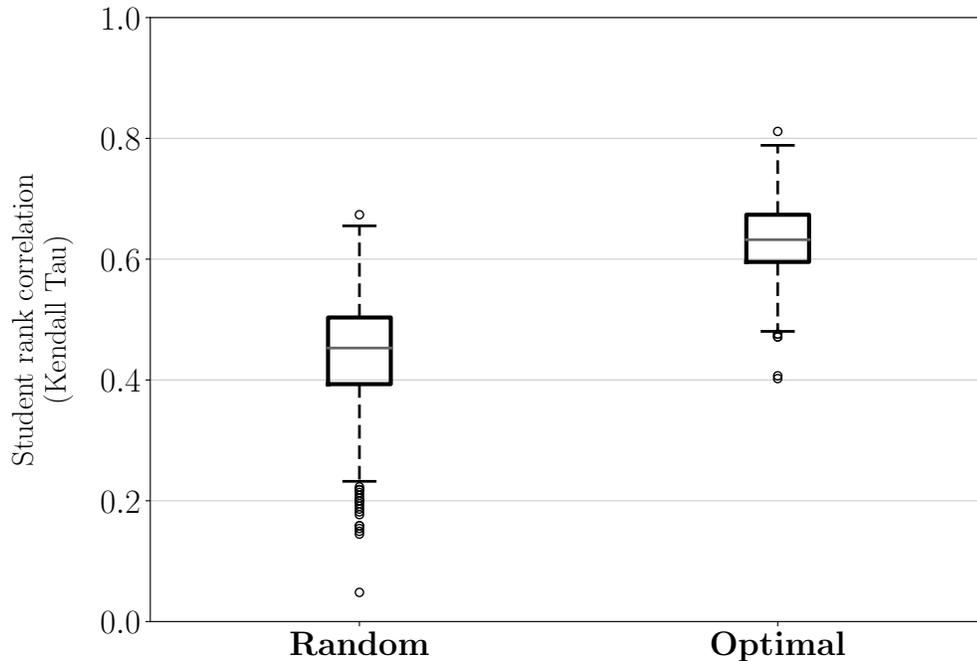


Figure 4.5: Rank correlation between workers ranked according to the raw scores attained on the **subsetMCQ** and **fullMCQ** tests, separated by choice set design strategy – choice sets optimized according to the proposed objective yield better rank correlation than choice sets sampled randomly.

the worker’s knowledge should result in a higher quality discrimination across workers of different levels of knowledge (abilities), and thus yield a more accurate ranking of the workers. We obtain a ranking of workers by sorting everyone according to their raw score on the **subsetMCQ**, and as in the within-subject analysis, evaluate it against the “ground-truth” ranking obtained by ordering the students by their raw score on the **fullMCQ** test. We compute rank correlation by sampling a random set of 50 workers and computing Kendall Tau for the **Random** and **Optimal** conditions, repeating the process for 1000 iterations and report the statistics in Figure 4.5.

We observe that rank correlation in the workers given a **subsetMCQ** test with the **Optimal** choice set significantly outperforms rank correlation of the workers

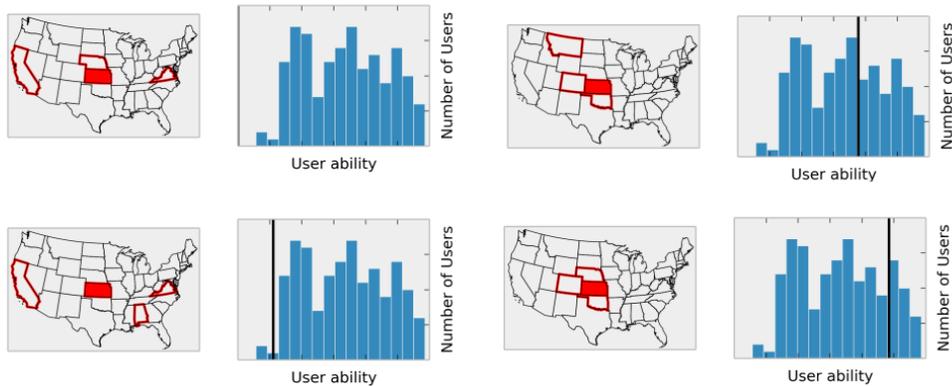


Figure 4.6: Visualization of optimal choice sets for the question “Kansas”, optimized to students of varying prior ability parameter (black vertical bar, displayed over the empirical distribution of inferred student abilities). Observe that as ability increases, choices become clustered closer to the true answer, making the correct answer more difficult to discern.

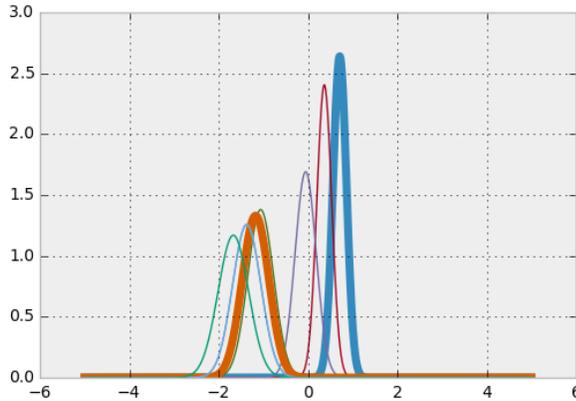
given a **subsetMCQ** test with a **Random** choice set ( $p$ -value=0 by permutation test), confirming our hypothesis: *a test that optimizes information about the student’s ability implicitly optimizes the accuracy of the ranking of the students.*

#### 4.10 Crowdsourcing tests from forums

One application that we explore in this paper is to the task of generating multiple choice tests from technical forum data. Technical forums, like StackExchange, Piazza and Quora, exhibit a typical structure: (i) a user posts a question on the forum, (ii) other users propose solutions by submitting answers, and (iii) users vote on what they consider to be the best answer to the original question. Forums that follow this structure provide an opportunity to apply our model for optimal

Physics #147346

I would like to know if it is possible to create a parachute so large in the real world that it might stop all velocity, essentially making whatever is attached to it float in mid-air

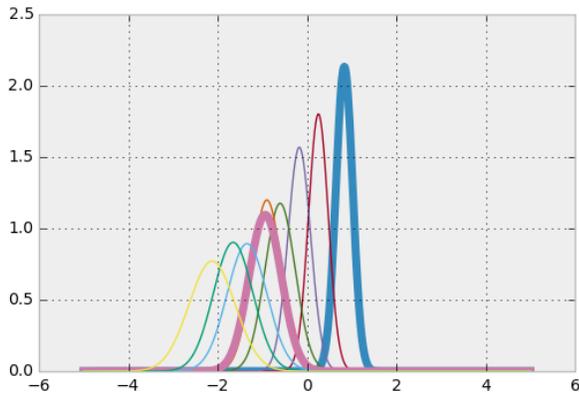


**A** No. All parachutes, whether they are drag-only (round) or airfoil (rectangular) will sink. Some airflow is needed to stay inflated, and that airflow comes from the steady descent. Whether your net descent rate is positive or negative is a different question. It is quite easy to be under a parachute and end up rising (I have done it myself), you just need an updraft in excess of your descent rate. Never lasts though, as a permanently floating parachute would violate a couple of laws of nature.

**B** It could be possible if the parachute was very large, rigid, shaped like a floating object, and you started descending from the vacuum of space. In this case the parachute would float on top of the atmosphere. It's easier to visualize if you imagine the parachute being a boat and you fell into some water; the boat would float on top of the water and reduce your velocity to zero.

Physics #776

Can we ignite Jupiter so that it will produce enough heat to warm these two earth like planets/satellites?

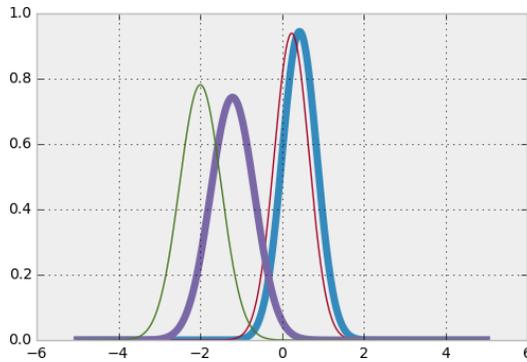


**A** Jupiter's mass is too small to produce nuclear fusion. Jupiter would need to be about 75 times as massive to fuse hydrogen and become a star. This wikipedia page explains the detailed requirements of nuclear fusion: [http://en.wikipedia.org/wiki/Nuclear\\_fusion](http://en.wikipedia.org/wiki/Nuclear_fusion)

**B** The most likely answer is that probably Jupiter is already ignited: it emits lots of infrared radiation that as far as i know, its largely unexplained. It just doesn't have enough mass to radiate more energy than a extremely dim brown dwarf, so it naively looks to us as a planet

Figure 4.7: Example StackExchange questions with posterior distributions over choice correctness parameters. Two optimal choices are highlighted and annotated. See Section 4.10.3 for details.

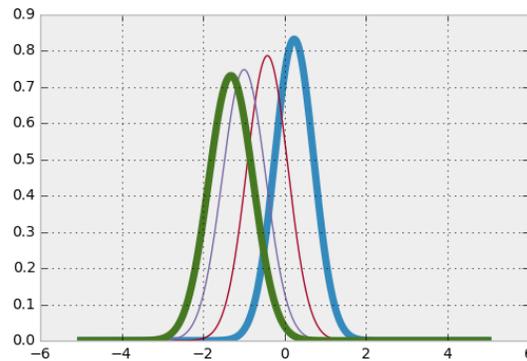
Physics #14609  
 Is there any difference between using  
 a positive versus a negative  
 charge to test an electric field?



**A** You can use a negative charge to test an electric field. You just have to remember that the electric field points antiparallel (opposite) to the force on the charge, rather than parallel to it (in the same direction). That's just a convention, though; we could have defined the electric field to point with the force on a negative charge, and physics would work the same, except for a couple of negative signs in some formulas.

**B** We take positive charge as a test charge because positive charge is higher potential and negative charge is lower potential. Therefore, influence of positive charge on other charges is greater than negative charges. We can also take negative charge but the effect will be lower.

Biology #8996  
 Does DNA have any other function  
 in the cell other than being  
 a genetic material and  
 carrier of information?



**A** In eukaryotes DNA has a structural as well coding function. Parts of chromosomes called centromeres bind to proteins and form a scaffold which helps chromosomes attach to each other and correctly segregate during division. Technically this is still related to the transmission of genetic information though...

**B** DNA has been shown to be important for biofilm formation in certain bacteria. This is extracellular DNA that comes from cell lysis.

Figure 4.8: Example StackExchange questions with posterior distributions over choice correctness parameters. Two optimal choices are highlighted and annotated. See Section 4.10.3 for details.

question generation where choice subsets are selected from the user submissions. The potential benefit of creating assessment content dynamically from technical forums is:

1. Large technical forums like StackExchange are repositories of real-world problems and solutions, where the solutions are of varying correctness and quality. A test generated from this data is likely to consist of relevant real-world problems.
2. Choices created from real user submissions are likely to capture common misconceptions that other people are likely to share and thus, are potentially good distractors.
3. The scope of sites like StackExchange could potentially facilitate test-generation customized to narrow target domains and subdomains of interest in areas for which testing material does not readily exist. One, for example, might want a test in a specific area of Parenting or a test that combines together multiple specific domains in programming.

#### **4.10.1 Modeling users and questions**

We describe how we adapt our model to the setting of a generic technical online forum that fits the structure described above, i.e., it contains user-submitted questions, user-submitted answers and user votes for each answer. Exactly as in the problem of “U.S. States Quiz,” we endow each choice (answer post) with a real-valued parameter  $\beta_{ij}$ , but where in this case  $i$  is an index of the user that contributed that answer and  $j$  is an index of the question which this answer answers. For modeling convenience, we explicitly distinguish between users

that contribute an answer, and users that vote for a particular answer. “Voting users” are modeled the same way as the users who answer multiple choice questions in our discrete choice model, i.e., their strictly positive “choosing ability”  $\theta$  appears as a coefficient of the choice correctness in parametrizing the discrete distribution over choices<sup>4</sup>. “Contributing users” are endowed with an “answering ability” parameter  $\phi$ , which parameterizes the distribution over answer correctness parameters for answers contributed by that user. This allows us to share statistical strength of “good” and “bad” answers that are created by the same users, e.g., users that contribute poor answers in general (answers that receive few upvotes) will be informative in inferring answer parameters in other questions they answered, where the voting information may be sparse.

#### 4.10.2 Generative Model and Inference

We formalize the above model with a Bayesian generative story shown on the right. We put normal priors on the answer and user parameters, and a truncated-normal prior on the voter ability, to ensure non-negativity. The high-level description of the story is as follows: users with ability  $\phi_i$  contribute answers to questions whose correctness  $\beta_{ij}$  is normally distributed about the creator’s ability, i.e., more able users are able to create higher-quality answers. Later at some time  $t$ , a voter with ability  $\theta_k$  observes a set of answers  $C_q^t$  (to question  $q$ ) that have been created up to time  $t$  and makes a selection according to the discrete distribution parametrized by (4.1), where voters with greater ability are more likely to pick the best choice. We use variational message passing for inference, a deterministic approximate posterior inference algorithm, provided in the In-

---

<sup>4</sup>unfortunately StackExchange datasets do not reveal the identity of the “voters”, thus we assume that each vote is contributed by a distinct “voter”

fer.NET package [127]. We perform inference on three StackExchange forums: *Biology* (620 users, 638 questions), *Physics* (3,487 users, 5,262 questions), *Parenting* (1,820 users, 1,503 questions).

**For each** user  $i \in S$ :

– Draw user ability  $\phi_i \sim \mathcal{N}(0, \sigma_{prior}^2)$

**For each** answer  $j$  created by user  $i$ :

\* Draw  $\beta_{ij} \sim \mathcal{N}(\phi_i, \sigma_{prior}^2)$

Draw  $\mu_\theta \sim \text{TruncNormal}(0, \sigma_{prior}^2)$

Draw  $\sigma_\theta^2 \sim \text{Inv-Gamma}(\alpha_{prior}, \beta_{prior})$

**For each** question  $q$ :

– **For each** vote in question  $q$  at time  $t$

\* Draw voter ability  $\theta_{qk} \sim \mathcal{N}(\mu_\theta, \sigma_\theta^2)$

\* Draw vote  $z_{qk} \sim \text{Discrete} \left( \{\pi_{qk}^{(i,j)}\}_{(i,j) \in C_q^t} \right)$  where  $C_q^t$  is a set of answers available for question  $q$  at time  $t$  and

$$\pi_{qk}^{(i,j)} = \frac{\exp(\theta_{qk} \beta_{ij})}{\sum_{(i',j') \in C_q^t} \exp(\theta_{qk} \beta_{i'j'})}$$

### 4.10.3 Examples

We present a qualitative analysis of the results via examples in Figure 4.7, which provide some insight into the advantages and issues with applying our model to real-world forum data at the task of question generation. Full end-to-end

evaluation of the quality and effectiveness of the generated questions will require user-studies, which we leave for future work. Figure 4.7 displays posteriors over answer correctness parameters for four questions, with the highlighted and annotated answers belonging to the optimal choice set, where optimality is determined by the optimality criterion introduced in Section 4.5. As done in Section 4.8.1, we optimize the choice sets for an “average user,” i.e., whose ability is given by the posterior mean of  $\theta$ . Finally, in selecting choice pairs, we require that the “most correct” choice (one with the highest posterior mean) always appears in the set, making the selection problem essentially one of finding a good distractor.

The examples in Figure 4.7 are given with their respective forum name and a question ID, and can be viewed in more detail by finding them on the Stack-Exchange site. For example, the top left question in Figure 4.7 (147346), can be found at: <http://physics.stackexchange.com/questions/147346>. Questions 14736, 14609 and 776 are examples where the distractors are all plausible incorrect answers (the correct answer in every question is marked with “A”). Question 8996, however, is a common example of a generated choice set, where the distractor is also a correct answer, yet it appeared less popular for another reason, e.g., it was incomplete, had little supporting evidence, or was simply not a commonly-known answer (the case for question 8996) and therefore received significantly fewer votes. In our setting, we argue that having an explicit constraint that the distractor is wrong is not necessary—it is sufficient if the user can tell apart the best answer from the remaining answers. However, if the dimension of quality is orthogonal to correctness, e.g., if one of the answers is better phrased or contains additional illustrations, the question will not serve its purpose in differentiating those users that know the answer from those that

do not. This limitation is potentially less severe in areas where the answer is constrained to be of a particular format, e.g., if the answer is computer code like in StackOverflow, where often multiple submitted answers may be correct, but only one exhibits the best performance. We leave the full study of the application of this model to test generation from technical forums for future work.

#### **4.11 Discussion**

We have proposed a method for optimal choice selection for the task of optimal test design. Our response model is closely related to a discrete choice model, where the variance parameter encodes the ability of the user. This formulation, unlike related models such as [53, 7], allows us to explicitly identify optimal choice sets, where optimality is specified in terms of estimator efficiency on the user ability parameter. We have demonstrated that the resulting choice sets are selected on the basis of how easily the choices are mistaken for one another, highlighting one of the principles of multiple choice question design: *good distractors must capture common misconceptions*. We also look ahead to the application of this model to data-driven crowd-sourced assessment generation from technical forums, and briefly highlight challenges and potentials of this paradigm.

#### **4.12 Future Work**

We focus on the limitations and extensions of the current model to use-cases where users interactively submit answers, which are then re-used as choices in

a multiple choice test, similar to the technical forum setting described in the previous section. The application scenario may be a large MOOC-like classroom, where hundreds of students submit answers to a question, and these answers are to be re-used either for the same or the next batch of students in the form of a well-designed multiple choice question. Two important challenges arise in this setting:

1. **Independence of irrelevant alternatives:** Many students are likely to submit similar answers: e.g., lexical, syntactic, or semantic variations. In such cases, the assumption that choices are made independently is violated. If the same optimization objective were to be used to select optimal choices when some of the choices are identical, the resulting choice sets would likely feature such identical choices. A model that explicitly clusters or models the correlation structure between choices is needed. We will address this problem in Chapter 6.
2. **Exploitation vs. exploration of choices:** In a big classroom where new answers arrive constantly, it is important that in addition to optimizing the choice set to best learn about the user, the model learns about the choice parameters themselves. The right system must balance between exploiting the choices that are optimal for learning about the user, and choices whose parameters are uncertain.

Finally, we believe that the most promising application of models such as the one proposed here will be in the area of data-driven test design, where the data is in the form of existing questions and answers, found in technical forums. As such places on the web can only be expected to grow in the immediate future, so will the need to leverage real-life problems and solutions to create relevant,

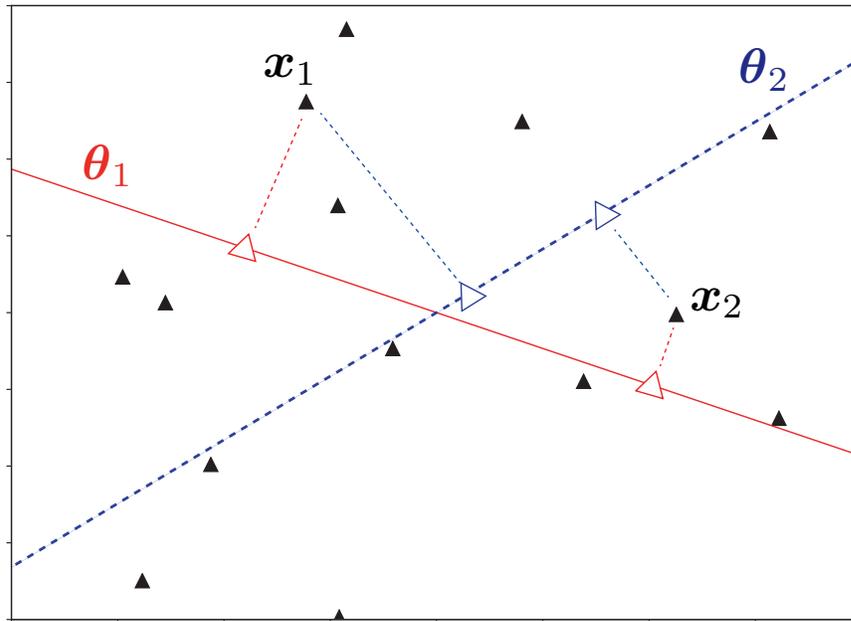


Figure 4.9: An illustration of a two-dimensional embedding of items (e.g., movies), represented by black triangles, and users, represented by hyperplanes  $\theta_1$  and  $\theta_2$ . A user's relative preference towards items is represented by a projection of the item's vectors  $x$  onto the user hyperplane  $\theta$ , i.e.,  $\theta^T x$  (hollow triangles). The task of optimal choice set design is to select a cardinality-constrained subset of the items to most efficiently learn the user's preference parameters  $\theta$ .

real-world tests in the rapidly growing and evolving technical domains. In the next chapter, we focus on the task of assessment (grading and ranking of students) based on their responses to multiple choice questions constructed from open-response submissions of other students.

### 4.12.1 Multidimensional extension

In this section, we briefly consider a multidimensional extension of the scalar model (4.1) proposed in this chapter. We show that the optimization problem that we derived for the scalar model can be naturally extended to a multidimensional setting. Potential applications extend outside education and include finding optimal choice sets (of movies, songs, images, etc.) for querying users in order to most efficiently learn about their preferences (e.g., in a cold-start regime of a recommender system). In this section, we derive an optimization objective in a multidimensional setting and demonstrate the potential of active choice selection for the task of item recommendation via simulations only.

A natural generalization of the model in (4.1) is to allow both users and choices to be represented as vectors in a  $D$ -dimensional space:

$$P(\text{user } i \text{ picks option } j \mid \boldsymbol{\theta}_i, \{\boldsymbol{x}_j\}_{j \in C}) = \frac{\exp(\boldsymbol{\theta}_i^T \boldsymbol{x}_j)}{\sum_{j' \in C} \exp(\boldsymbol{\theta}_i^T \boldsymbol{x}_{j'})} \quad (4.7)$$

where  $\boldsymbol{\theta}_i \in \mathbb{R}^D$  and  $\boldsymbol{x}_j \in \mathbb{R}^D$  are the user and choice parameters respectively. In the context of choice-making, this model is known as a Multinomial logit (MLN) discrete choice model [121], which specifies the likelihood of a user making a selection out of a set of options based on the attributes of the options (given in  $\boldsymbol{x}_j$ ) and the user's preferences (given in  $\boldsymbol{\theta}_i$ ). Recall that in our original scalar model given in (4.1), the user's scalar parameter  $\theta$  was interpreted as "ability" as a consequence of interpreting the item (choice) with the maximum parameter value as a "correct answer". While this interpretation no longer holds when the user and item parameters are generalized to be multidimensional, the optimization problem introduced for finding optimal choice sets in the scalar case can nevertheless be generalized to a multidimensional setting. In a multidimensional setting, optimal choice sets that maximize information about the user parameter

$\theta$  would be choice sets that are particularly “revealing” of the user’s preferences, rather than the user’s ability (as was our original interpretation of optimal choice sets in the scalar setting). We can show (Section 4.13.2) that the optimization problem for finding an optimal choice set in a multidimensional setting is given by:

$$\begin{aligned}
& \underset{\{x_n\}}{\text{maximize}} && \frac{\sum_i^N \sum_{j>i}^N x_i x_j \|\mathbf{x}_i - \mathbf{x}_j\|_2^2 \exp(\boldsymbol{\theta}^T [\mathbf{x}_i + \mathbf{x}_j])}{\sum_i^N \sum_j^N x_i x_j \exp(\boldsymbol{\theta}^T [\mathbf{x}_i + \mathbf{x}_j])} \\
& \text{subject to} && x_n \in \{0, 1\}, \forall n \in Q \\
& && \sum^N x_n \leq K
\end{aligned} \tag{4.8}$$

where by abuse of notation,  $x_i$  and  $x_j$  are binary variables indicating which choices are included in the choice set and  $\mathbf{x}_i$  and  $\mathbf{x}_j$  are vector-valued parameters describing the choices. Note that the multidimensional formulation above generalizes the scalar optimization problem given in (4.11) by replacing squared differences between scalar-valued choice parameters with squared Euclidean distances between vector-valued choice parameters.

## Synthetic experiments

We conduct a simulation study to (i) evaluate the efficacy of the proposed optimal choice set selection algorithm and (ii) gain understanding into the effect of the number of choices (in a choice set) on the performance gains. In this simulation study, we consider that there is a single user and a collection of 50 items (e.g., movies), represented by the parameter vectors  $\boldsymbol{\theta}$  and  $\{\mathbf{x}_j\}$  respectively. All parameter vectors are sampled from an independent  $D$ -dimensional multivariate normal distribution ( $D = 10$  in our simulations) with mean  $\mathbf{0}$  and covariance  $2I$ .

For evaluation, we consider predicting the user’s top choice among a subset

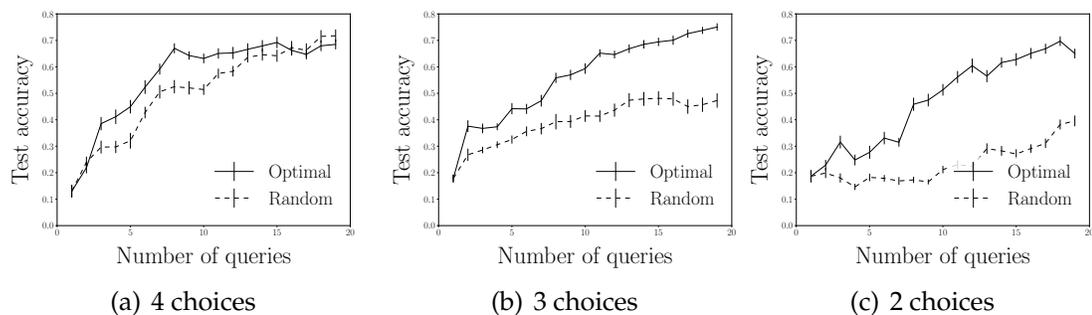


Figure 4.10: Accuracy in predicting users’ held-out preferences as a function of the number of queries made to a user in the training session (simulation). Figures (a), (b) and (c) simulate settings where the choice sets consist of 4, 3 and 2 options respectively. Choice sets are generated (i) by solving the optimization problem in (4.8) and (ii) by sampling choices uniformly at random. Optimal choice sets outperform randomly generated choice sets, however, the gains are especially large when the number of choices shown is small.

of items, where an instance consists of a subset of 20 items (out of 50 items) and 60 such instances comprise the evaluation set. Training the model involves estimating the user parameter  $\theta$  via a sequence of a fixed number of queries, where a query is a multiple choice question, requesting the user to select their top choice among the subset of presented choices (we experiment with choice sets of size 2, 3 and 4). We evaluate two methods for constructing the choice sets: (i) optimally by solving (4.8) where we use the model’s current ML estimate of  $\theta$  at a every interaction and (ii) randomly, by sampling a subset of choices uniformly at random. Our evaluation metric is accuracy in predicting the user’s top choice on the set of instances in the evaluation set (described at the beginning of this paragraph). We report average accuracy, averaged across 10 simulation runs (where each run involves sampling a new set of user and item parameters).

Figure 4.10 depicts accuracy for both (i) optimal and (ii) random strategies in generating choice sets, partitioned by the size of the choice set. On the basis of

these results, we can conclude the following: (i) the optimal strategy outperforms a random strategy regardless of the number of choices in a choice set, however (ii) the optimal strategy yields a greater performance gain over random for smaller choice sets. This observation agrees with the intuition that larger choice sets are more informative about the user than smaller choice sets (as the user reveals more information about their preferences from one choice in a larger choice set), and that in the limit of showing the user all items in the collection as choices, there would be no difference between the two strategies.

A potential direction for future research is to develop optimal “quizzes” that are administered during a cold-start initialization of a recommender engine, e.g., a system that asks a user to answer a few multiple choice questions about his or her movie preferences, or intermittently during the lifetime of a recommendation system whenever it deems that a particular question would be informative in updating its model of the user. There has been work in addressing the cold-start problem in recommender systems via an “interview process”, where users are asked a sequence of questions about what movies they like and dislike [63, 211, 183]. Using multiple choice questions, where each user expresses relative preference, may provide a richer signal for bootstrapping such systems. We leave this line of inquiry for future research.

## 4.13 Appendix

### 4.13.1 Derivation of the optimal choice set (scalar case)

We formulate the optimization problem for choosing optimal choice sets by computing the Fisher information w.r.t.  $\theta$  as a function of the included choices (choice set) and maximizing this function. We can express the likelihood of student with ability  $\theta$  choosing the  $k^{\text{th}}$  option from set  $C$  as follows:

$$P(z = k \mid \{\beta_k\}_{k \in C}, \theta) = f(\theta; C) = \frac{\prod_k^K \exp(\theta\beta_k)^{\delta(z=k)}}{\sum_{k'}^K \exp(\theta\beta_{k'})}, \quad (4.9)$$

where  $k$  indexes the elements of set  $C$ . We can express the Fisher information of set  $C$  w.r.t. parameter  $\theta$  as follows:

$$\begin{aligned} \mathcal{I}(\theta; C) &= -\mathbb{E} \left[ \frac{\partial^2}{\partial \theta^2} \log f(\theta; C) \middle| \theta \right] \\ &= -\mathbb{E} \left[ \frac{\partial}{\partial \theta} \left( \sum_k^K \delta(z=k)\beta_k - \frac{\sum_k^K \beta_k \exp(\theta\beta_k)}{\sum_k^K \exp(\theta\beta_k)} \right) \middle| \theta \right] \\ &= \frac{\sum_k^K \beta_k^2 \exp(\theta\beta_k)}{\sum_k^K \exp(\theta\beta_k)} - \frac{\left( \sum_k^K \beta_k \exp(\theta\beta_k) \right)^2}{\left( \sum_k^K \exp(\theta\beta_k) \right)^2} \\ &= \frac{\sum_k^K \exp(\theta\beta_k) \sum_k^K \beta_k^2 \exp(\theta\beta_k)}{\left( \sum_k^K \exp(\theta\beta_k) \right)^2} \\ &\quad - \frac{\left( \sum_k^K \beta_k \exp(\theta\beta_k) \right)^2}{\left( \sum_k^K \exp(\theta\beta_k) \right)^2} \\ &= \frac{\sum_k^K \sum_{k'}^K \beta_k^2 \exp(\theta[\beta_k + \beta_{k'}])}{\sum_k^K \sum_{k'}^K \exp(\theta[\beta_k + \beta_{k'}])} \\ &\quad - \frac{\sum_k^K \sum_{k'}^K \beta_k \beta_{k'} \exp(\theta[\beta_k + \beta_{k'}])}{\sum_k^K \sum_{k'}^K \exp(\theta[\beta_k + \beta_{k'}])} \end{aligned} \quad (4.10)$$

$$\begin{aligned}
&= \frac{\sum_k^K \sum_{k' > k}^K (\beta_k^2 + \beta_{k'}^2 - 2\beta_k \beta_{k'}) \exp(\theta[\beta_k + \beta_{k'}])}{\sum_k^K \sum_{k'}^K \exp(\theta[\beta_k + \beta_{k'}])} \\
&= \frac{\sum_k^K \sum_{k' > k}^K (\beta_k - \beta_{k'})^2 \exp(\theta[\beta_k + \beta_{k'}])}{\sum_k^K \sum_{k'}^K \exp(\theta[\beta_k + \beta_{k'}])}.
\end{aligned}$$

If we let  $x_n \in \{0, 1\}$  be the decision variables over choices from  $Q$ , in the setting where we are looking to present only a limited number of choices ( $K$ ), we obtain the following optimization problem:

$$\begin{aligned}
&\underset{\{x_n\}}{\text{maximize}} && \frac{\sum_i^N \sum_{j > i}^N x_i x_j (\beta_i - \beta_j)^2 \exp(\theta[\beta_i + \beta_j])}{\sum_i^N \sum_j^N x_i x_j \exp(\theta[\beta_i + \beta_j])} && (4.11) \\
&\text{subject to} && x_n \in \{0, 1\}, \forall n \in Q \\
&&& \sum_{n=1}^N x_n \leq K
\end{aligned}$$

### 4.13.2 Derivation of the optimal choice set (multidimensional case)

We extend the derivation in Section 4.13.1 to a multidimensional setting. Consider that users and choice items are embedded in a  $D$ -dimensional space where  $\boldsymbol{\theta} \in \mathbb{R}^D$ , and  $\mathbf{x} \in \mathbb{R}^D$  represent user and choice parameters respectively. Similar to (4.9), we can express the likelihood of a user selecting choice  $k$  out of set of choices  $C$  in a multidimensional setting as follows:

$$P(z = k \mid \{\mathbf{x}_k\}_{k \in C}, \boldsymbol{\theta}) = f(\boldsymbol{\theta}; C) = \frac{\prod_k^K \exp(\boldsymbol{\theta}^T \mathbf{x}_k)^{\delta(z=k)}}{\sum_{k'}^K \exp(\boldsymbol{\theta}^T \mathbf{x}_{k'})}, \quad (4.12)$$

where  $k$  indexes the elements of set  $C$ . We can express the Fisher information of set  $C$  w.r.t. parameter vector  $\boldsymbol{\theta}$  as follows:

$$\mathcal{I}(\boldsymbol{\theta}; C) = - \left( \mathbb{E} \left[ \frac{\partial^2}{\partial \theta^{(i)} \partial \theta^{(j)}} \log f(\boldsymbol{\theta}; C) \mid \boldsymbol{\theta} \right] \right)$$

where  $\mathcal{I}(\boldsymbol{\theta}; C)$  is a Fisher information matrix. Minimizing the sum of the ML estimator variance of each component of  $\boldsymbol{\theta}$  can be achieved by minimizing the trace of the Fisher information matrix:

$$\begin{aligned}
\text{Tr}(\mathcal{I}(\boldsymbol{\theta}; C)) &= -\text{Tr}\left(\mathbb{E}\left[\frac{\partial^2}{\partial\theta^{(i)}\partial\theta^{(j)}}\log f(\boldsymbol{\theta}; C)\middle|\boldsymbol{\theta}\right]\right) \\
&= -\sum_i^D\mathbb{E}\left[\frac{\partial}{\partial\theta^{(i)}}\left(\sum_k^K\delta(z=k)x_k^{(i)}-\frac{\sum_k^Kx_k^{(i)}\exp(\boldsymbol{\theta}^T\mathbf{x}_k)}{\sum_k^K\exp(\boldsymbol{\theta}^T\mathbf{x}_k)}\right)\middle|\theta^{(i)}\right] \\
&= \sum_i^D\frac{\sum_k^K(x_k^{(i)})^2\exp(\boldsymbol{\theta}^T\mathbf{x}_k)}{\sum_k^K\exp(\boldsymbol{\theta}^T\mathbf{x}_k)}-\frac{\left(\sum_k^Kx_k^{(i)}\exp(\boldsymbol{\theta}^T\mathbf{x}_k)\right)^2}{\left(\sum_k^K\exp(\boldsymbol{\theta}^T\mathbf{x}_k)\right)^2} \\
&= \sum_i^D\frac{\sum_k^K\sum_{k'>k}^K(x_k^{(i)}-x_{k'}^{(i)})^2\exp(\boldsymbol{\theta}^T[\mathbf{x}_k+\mathbf{x}_{k'}])}{\sum_k^K\sum_{k'}^K\exp(\boldsymbol{\theta}^T[\mathbf{x}_k+\mathbf{x}_{k'}])}. \\
&= \frac{\sum_k^K\sum_{k'>k}^K\|\mathbf{x}_k-\mathbf{x}_{k'}\|_2^2\exp(\boldsymbol{\theta}^T[\mathbf{x}_k+\mathbf{x}_{k'}])}{\sum_k^K\sum_{k'}^K\exp(\boldsymbol{\theta}^T[\mathbf{x}_k+\mathbf{x}_{k'}])}.
\end{aligned}$$

where between steps 3 and 4 of the above, we relied on the same set of algebraic manipulations as done in (4.10). We use the notation  $x^{(i)}$  to represent the  $i^{\text{th}}$  component of the  $D$ -dimensional vector  $\mathbf{x}$  (and similarly for  $\boldsymbol{\theta}$ ). The resulting optimization problem generalizes (4.10), where the squared difference between each scalar choice parameter is subsumed by a Euclidean distance between the parameter vectors.

### 4.13.3 Optimal choice set when $\theta = 0$

We have shown that in the limit of a user with low ability, the optimal choice set is obtained by maximizing:

$$\mathcal{I}(\boldsymbol{\theta}; C) = \frac{1}{K^2}\sum_k^K\sum_{k'>k}^K(\beta_k-\beta_{k'})^2$$

To investigate the placement of choices that maximizes above, without loss of generality, consider that all choice parameters are constrained to a unit interval,

i.e.,  $0 \leq \beta_k \leq 1, \forall k$ .

**Claim 3.** *The information on  $\theta$  is maximized when half of the choices are placed at one end of the interval, and half of the choices are placed at the other end.*

*Proof.* Consider an arbitrary placement of  $\{\beta_k\}$  on a unit interval. Pick any  $\beta_k$ . Moving  $\beta_k$  to either end of the interval will guarantee an improvement of the objective. To see that let  $x = \beta_k$ . We are then interested in finding  $x$  that maximizes (ignoring terms that do not depend on  $x$ ):

$$\sum_k^{K-1} (\beta_k - x)^2$$

This is a quadratic function with a minimum at  $x_{min} = \frac{\sum_k^{K-1} \beta_k}{K-1}$ . Since  $0 \leq \beta_k \leq 1$ , we have that  $0 \leq x_{min} \leq 1$ , i.e., regardless of the placement of the other choices, the objective is always improved by moving  $\beta_k$  to one of the endpoints. Suppose that we place  $n_0$  choices at the  $\beta = 0$  endpoint and  $n_1$  choices at the  $\beta = 1$  endpoint. The objective value is then given by  $\min(n_0, n_1)$ , which is maximized when half of the choices are placed on one end and half of the choices on the other (if number of choices is odd, placement of one choice is arbitrary).  $\square$

For a given set of  $\{\beta_k\}$  we have the following characterization of an optimal solution.

**Claim 4.** *Let  $\beta_1 \leq \beta_2 \leq \dots \leq \beta_n$ . For any  $\ell$  that is in the optimal choice set, we may assume that either  $\ell - 1$  or  $\ell + 1$  is also in the optimal choice set (or  $\ell = 1$  or  $\ell = n$ ).*

*Proof.* Assume by means of contradiction that  $\ell$  (not equal to 1 or  $n$ ) is in the optimal choice set, denoted by  $C$ , while both  $\ell - 1$  and  $\ell + 1$  are not. Let  $m$  be an

element not in  $C$  (which will be one of  $\ell - 1$  or  $\ell + 1$ ). Consider the difference in objective value for the solutions  $C$  and  $C \setminus \{\ell\} \cup \{m\}$ .

$$\begin{aligned}
& \sum_{i,j \in C \setminus \{\ell\} \cup \{m\}} (\beta_i - \beta_j)^2 - \sum_{i,j \in C} (\beta_i - \beta_j)^2 \\
&= \sum_{i \in C \setminus \{\ell\}} (\beta_i - \beta_m)^2 - \sum_{i \in C} (\beta_i - \beta_\ell)^2 \\
&= \sum_{i \in C} (\beta_i - \beta_m)^2 - (\beta_\ell - \beta_m)^2 - \sum_{i \in C} (\beta_i - \beta_\ell)^2 \\
&= \sum_{i \in C} (\beta_i - \beta_\ell + \beta_\ell - \beta_m)^2 - (\beta_\ell - \beta_m)^2 - \sum_{i \in C} (\beta_i - \beta_\ell)^2 \\
&= \sum_{i \in C} \left( (\beta_i - \beta_\ell)^2 + (\beta_\ell - \beta_m)^2 + 2(\beta_i - \beta_\ell)(\beta_\ell - \beta_m) \right) \\
&\quad - (\beta_\ell - \beta_m)^2 - \sum_{i \in C} (\beta_i - \beta_\ell)^2 \\
&= (|C| - 1)(\beta_\ell - \beta_m)^2 + 2(\beta_\ell - \beta_m) \sum_{i \in C} (\beta_i - \beta_\ell).
\end{aligned}$$

Note that the first term is always nonnegative, and the second term has different signs for  $m = \ell - 1$  and  $m = \ell + 1$ . Therefore swapping out  $\ell$  for either  $\ell - 1$  or  $\ell + 1$  does not decrease the objective value.  $\square$

**Corollary 2.** *There exists an algorithm that runs in  $O(n^2)$  time to find an optimal solution to (4.4) for the special case where  $\theta = 0$ .*

## CHAPTER 5

### JOINT ASSESSMENT AND GRADING

#### 5.1 Introduction

In the previous chapter, we focused on the problem of constructing multiple choice questions by optimally selecting choice sets out of a potentially large pool of available distractors. While the focus of the previous chapter was not on the source of these distractors, we alluded to the potential of crowdsourcing distractors from the students' own open-response solutions. In this chapter, we focus on the problem of summative assessment (e.g., grading and ranking students) via multiple choice questions whose options are crowdsourced from the open-response solutions submitted by students themselves. While this chapter will focus on the problem of open-response assessment and grading in the context of education, the presented framework, model and algorithms are general and can be applied in many crowdsourcing tasks where the goal is to identify quality contributions and competent contributors.

*Multiple-choice questions (MCQs)* are a common way to overcome the so-called *scaling problem* in assessment: grading a large number of submissions from many students in an efficient manner [158]. The immense scaling potential of MCQs to large classroom settings such as massive open online courses (MOOCs), however, comes at the cost of rigidity and considerable sensitivity to its design—questions and options that are designed without the consideration of the students' likely misconceptions, for example, can yield uninformative questions [67, 157] (e.g., when the distractors are easy to eliminate). As a consequence, the effort saved in grading is often offset by the effort required to design an effective MCQ test.

More recently, the practical realization of MOOCs opened another opportunity to solving the problem of scaling in large-scale assessments, one that leverages the size of the classroom to its advantage: *peer-grading*. In its traditional form, peer-grading assigns to each student a secondary role of a grader. Students are responsible for validating the correctness of other students' solutions in order to assign a score, typically in accordance with a rubric provided by an instructor. The advantage of peer-grading is its flexibility to students' submissions, which may range from short answers, to essays, diagrams, code, or entire projects [95]. In a practical deployment, however, peer-grading, faces the same challenges as crowdsourcing. First, each student differs in their ability to grade, and the grades assigned by different students must be reconciled in a reasonable way. Second, grading becomes an additional burden on the students, and mechanisms must be put in place that not only incentivize participation and effort, but prevent students from "gaming" the process. Recent research in peer-grading has started to address these challenges [143, 148, 203]

### **5.1.1 JAG: An alternative to Peer Grading**

We present a novel alternative approach to peer-grading that naturally resolves the challenges of grade aggregation and incentivization. We propose *joint assessment and grading (JAG)*, which fuses grading and assessment into a single, streamlined process by re-framing grading as additional testing. Our approach is motivated by the fact that a "grader" that has no answer key, when presented with the listing of other students' answers, is no different than a test-taker facing a multiple-choice question (with multiple possible correct or incorrect answers). In other words: a student selecting what they believe to be the correct answer in

a multiple choice question (MCQ) constructed from the open-response submissions of other students is in effect simultaneously (i) grading the other students and (ii) being assessed by his or her ability to select the correct answer. In peer-grading, we already face the challenge of noisy inputs from the (potentially unmotivated) graders. By re-framing the act of grading as that of MCQ testing, the source of the apparent noise in grading becomes distributed according to the ability of the students in the class.

The proposed mechanism of JAG combines the advantages of both worlds: the structure of multiple choice questions and the flexibility to general response types offered by *peer-grading*. First, by constructing the MCQs directly from students' open-response submissions, the questions naturally capture the distribution of misconceptions present in the population of students being tested, requiring little to no instructor input. Second, our framework offers a mechanism for automatically grading open-response submissions, thus facilitating greater student engagement and higher-order thinking characteristic to open-response questions [67]. Third, by re-framing the task of grading as that of testing, the students are incentivized in the context of a familiar task: namely by expending their effort towards correctly answering an MCQ, they are implicitly directing that effort towards grading other students' submissions. At the same time, the students are not burdened with (what they may perceive as) a "thankless" job of grading, but instead in the process of answering the additional MCQs, the students are provided with an additional opportunity to demonstrate their knowledge.

In this chapter, we formalize the process of JAG as a statistical estimation problem. At the heart of our approach is the traditional Rasch model that captures

the interaction between student abilities and question difficulties in determining the likelihood of a student answering a question correctly [149]. We develop an expectation maximization (EM) algorithm for estimating the parameters of the proposed model in an unsupervised setting (i.e., in absence of an answer key), and demonstrate the effectiveness of our framework through a real-world user-study conducted on Amazon’s Mechanical Turk. Additionally, we investigate the key properties and limitations of our approach via simulations.

## 5.2 Related Work

The work in this chapter builds on the recent progress in two distinct areas: *crowd-sourcing* and *peer-grading*, that we unite and extend within our proposed framework for *joint assessment and grading (JAG)*.

### 5.2.1 Crowdsourcing

An important task in *crowdsourcing* is known as *label-aggregation*, and is concerned with the problem of optimally recovering some underlying ground truth (e.g., image class label) from a number of (unreliable) human judgements. See [82] for a detailed review. In the context of education, the task of automatically identifying the correct answers from open-response submissions is closely related to the task of label aggregation. Within the field of crowd-sourcing, the work of [42, 199, 7] are the most related to our approach. [42] was the first to suggest an expectation maximization (EM) algorithm for label aggregation, motivated by a clinical setting of making a diagnosis. More recently, [199] extended this

approach to model the variation in task difficulty in the context of image labeling. In the context of education, [7] has proposed a statistical model for aggregating answers from “noisy” students, with the goal of automatically identifying the correct answers to MCQs. They deploy an Expectation Propagation (EP) algorithm for Bayesian inference, and demonstrate the ability to infer correct answers accurately in a setting of an IQ test. The work in this chapter can be seen as a generalization of [199, 7], where we explicitly model the dependence among question choices and students that generate those choices in the context of answering open-response questions.

### 5.2.2 Peer-grading

Much of the recent research in *peer-grading* addresses a related problem of aggregating a number of “noisy” grades submitted by students in a statistically principled manner. Models such as the ones in [143, 148] pose the problem of peer-grading as that of statistical estimation. Since traditional grading assumes that graders are in possession of a grading rubric, statistical models of peer-grading are concerned primarily with accounting for the reliability and bias of graders in evaluating assignments against a gold-standard. In contrast to such “explicit” models of grading, we view grading as an implicit process that results as a by-product of students’ genuine attempt to answer MCQs constructed from the open-response submissions of other students. As such, we do not require additional “grader-specific” parameters, as grading in our framework is subsumed by the response model (model of how students answer questions as a function of their ability and question difficulty). We do note, however, that one of the proposed models in [143] explicitly couples grading and ability parameters

in an attempt to capture the intuition that better students may also be better graders. This intuition can be viewed as being taken to its extreme in our setting: removing the boundary between grading and test-tasking ensures that better students are more reliable graders by construction.

### 5.2.3 Clustering submissions

We also take note of an emerging area of work focused on clustering open-response submissions (not necessarily for peer-grading). An important by-product of scaling in the context of assessment is the inevitable increase in similarity between student open-response submissions, which introduces redundancy during grading. Moreover, a recent theoretical result of [175] indicates that without some way of reducing dimensionality of submissions, there will always be a constant number of misgraded assignments (assuming certain scaling properties of the classroom). In an effort to reduce the workload of the instructors (or peers), there has been a number of successful attempts to cluster responses in specific domains, e.g., language [10, 25] and mathematics [102]. Answer clustering is even more critical in the framework of JAG, where the practical constraints of testing limit the number of options that can be shown in a multiple choice question. To generate effective questions, the presented options must offer a representative sample of the diverse open-response submissions in a large classroom. In the next chapter, we demonstrate that the pattern of selected options alone provides the necessary signal to perform domain-agnostic clustering of submissions (i.e., without considering the content of the answers). This observation demonstrates the tremendous versatility of this framework to jointly assess, grade and cluster open-response submissions.

## 5.3 Model

### 5.3.1 Fully observed setting

We start by reviewing the classic IRT Rasch model that will serve as the foundation of our approach. Consider a set of students  $S$  and a set of questions  $Q$ , where a student  $i \in S$  is endowed with an ability parameter  $s_i \in \mathbb{R}$ , and each question  $j \in Q$  is endowed with a difficulty parameter  $q_j \in \mathbb{R}$  (note that we capitalize all sets in our notation). By abuse of notation, we will often overload  $s_i$  to refer to both, the student index  $i$  and their ability, depending on the context; the same applies to  $q_j$ , which we use to refer to the question itself as well as its difficulty. The well-established 1-PL IRT Rasch model [149] expresses the probability that the student  $s_i$  answers question  $q_j$  correctly via the following likelihood function:

$$P(z_{i,j} \mid s_i, q_j) = \frac{1}{1 + \exp(-z_{i,j}(s_i - q_j))}, \quad (5.1)$$

where  $z_{i,j} \in \{+1, -1\}$  is the binary outcome of student  $s_i$ 's attempt of question  $q_j$ ; we use +1 and -1 to designate correct and incorrect responses, respectively. If we are in the possession of an answer key for each question, then we also know  $\{z_{i,j}\}, \forall i, j$  (we will refer to this as the *fully observed* setting). This allows us to estimate the ability of each student and the difficulty of each question by maximizing the likelihood of all outcomes under our model:

$$\{s_i, \forall i, q_j, \forall j\} = \operatorname{argmax}_{s_i, q_i} \prod_{z_{i,j} \in D} P(z_{i,j} \mid s_i, q_j), \quad (5.2)$$

where  $D = \{z_{i,j}\}$  is the set of outcomes (e.g., of a test).

### 5.3.2 Partially observed setting

Consider now the setting where some (or all) of the outcomes  $z_{i,j} \in D$  are not observed. In practice, this is the case, for example, when the answer key to some of (or all) the questions is not available. In our setting, where the choices in the multiple choice question are in fact other students' submissions, the correctness of these submissions are not known a priori. Let  $A_j$  be the set of open-response answers submitted by a subset of students in  $S_{\text{open}} \subseteq S$  in response to the question  $q_j$ . At some later time, a student  $s_i \in S_{\text{mcq}} \subseteq S$  is presented with the same question  $q_j$ , but in the form of a multiple-choice question, with the options being exactly the answers in  $A_j$  (note that  $S_{\text{mcq}}$  need not be disjoint with  $S_{\text{open}}$ ). The student  $s_i$  is informed that there may be zero or more correct answers in the set of options in  $A_j$  and they are instructed to select "all that apply." The student  $s_i$  goes through each option in  $A_j$  and submits a response to that option. Let  $y_{i,j} = \{y_{i,j}^k\}$  be the set of such responses made by student  $s_i$  on the set of answers  $A_j$ , where  $y_{i,j}^k \in \{+1, -1\}$  is the student  $s_i$ 's selection on the  $k$ 'th answer (option) in  $A_j$ . In other words the variables  $y_{i,j}^k$  are the observations of whether the student  $s_i$  "clicked" on answer  $k$  to question  $j$  (i.e., that student judged that particular answer to be correct). In what follows, we describe the statistical model that relates the student and question parameters which we are interested in estimating, to the set of response observations. Our model consists of two components: (i) the *open-response* component that models the students (and their responses) that generate open-response answers, and (ii) the *multiple choice model* component that models the students (and their responses) that are presented with the multiple choice version of each question.

**Open-response model:** Because we do not know whether the submitted open-response answers are correct, we treat the correctness of each submission as a hidden variable  $z_{i,j} \in \{+1, -1\}$ ; this allows us to express the component of the overall likelihood of our data, responsible for the open-response answers only, as follows:

$$P(\{z_{i,j}\} \mid S_{\text{open}}, Q) = \prod_{z_{i,j}} P(z_{i,j} \mid s_i, q_j),$$

where  $P(z_{i,j} \mid s_i, q_j)$  is the Rasch likelihood given in (5.1). Note that we drop the  $k$ -superscript notation for the  $z_{i,j}$  variables because each student is assumed to provide at most one open-response submission to each question (since  $k$  indexes the answers to a specific question). The observed responses to the multiple-choice version of each question (described next) will provide the necessary data to estimate the parameters in the model, including the hidden variables  $z_{i,j}$ , i.e., the correctness of each open-response submission.

**Multiple choice model:** Now consider the setting where each question is presented in the form of an MCQ. Recall that a student answering a multiple choice question is presented with multiple options, each generated by some (other) student in the set  $S_{\text{open}}$ , and where several options (or even no options) may be correct. The intuition that we want to capture in our model is that a student of great relative ability (i.e.,  $s_i \gg q_j$ ) will select ( $y_{i,j}^k = +1$ ) the option (i.e., judge it as being correct) *if* that option is actually correct ( $z_j^k = +1$ ). The same student will *not* select that option ( $y_{i,j}^k = -1$ ) if that option is incorrect ( $z_j^k = -1$ ). At the same time, a student of poor relative ability (i.e.,  $s_i \ll q_j$ ) will not be able to identify the correct answer, regardless of whether the option is correct, i.e., they will guess. This intuition can be captured by the following function that

parametrizes the likelihood of student  $s_i$  selecting the option  $k$  to question  $q_j$ :

$$P(y_{i,j}^k | s_i, q_j, z_j^k) = \frac{1}{2} \left( \frac{1}{1 + \exp(-y_{i,j}^k z_j^k (s_i - q_j))} + 1 \right). \quad (5.3)$$

One can easily verify that this likelihood satisfies the requirements outlined above by considering every combination of the assignment to  $y_{i,j}^k$  and  $z_j^k$ , and taking the limits of  $s_i - q_j \rightarrow \infty$  (great relative ability) and  $q_i - s_j \rightarrow \infty$  (poor relative ability). Note that this time we drop the index  $i$  (index of the student who generated the option  $k$  in question  $q_j$ ) in  $z_j^k$ , as in the above, we use  $s_i$  to refer to the student answering the multiple choice version of the question. Note that the above likelihood follows the same intuition as proposed by [7], but in a setting with an arbitrary number of choices and one correct answer. In Figure 5.1 we illustrate both components of the likelihood (the *open-response* and the *multiple choice* component) as a graphical model. In this illustration, we use the notation  $s_{i'}$  to refer to the student that generated the answer and  $s_i$  to refer to the student that observes the answer as a choice in a multiple choice version of question  $q_j$ .

If we make a leap of assuming conditional independence between the student  $s_i^{\text{th}}$  responses to each option in a multiple choice question (conditional on  $s_i, q_j$  and  $z_j^k$ ), then we can express the likelihood of observing every response to every multiple choice question as follows:

$$P(\{Y_j\} | S_{\text{mcq}}, Q, \{Z_j\}) = \prod_{s_i \in S_{\text{mcq}}} \prod_{q_j \in Q} \prod_{\substack{y_{i,j}^k \in Y_j \\ z_j^k \in Z_j}} P(y_{i,j}^k | s_i, q_j, z_j^k).$$

The assumption of conditional independence requires some additional justification in our setting. Intuitively, we are justified in claiming conditional independence when we believe that the set of conditioning variables accounts for everything that may be shared across observations, such that the only remaining source of the variance is noise. For example, observations of different students

answering the same question on the test are conditionally independent given the difficulty of that question. In modeling the likelihood of a student selecting each option in a multiple choice question, however, we overlook the potential for the options to be related. In an extreme example, two options may be identical or paraphrases of each other, which we expect to be common-place when these options are generated by students in a large classroom. In this case, conditional independence no longer holds without an introduction of additional conditioning variables that group the related options in some way. To some extent, this problem can be mitigated by pre-processing and clustering similar answers before displaying them as options in a multiple choice question. This is a strategy that we take in this chapter. In the next chapter, we develop a method for automatically clustering answers based on the response patterns to multiple choice questions.

To complete our model, we combine the *open-response* and the *multiple-choice* components:

$$P(\mathbf{y}, \mathbf{z} \mid \mathbf{s}, \mathbf{q}) = \underbrace{P(\mathbf{y} \mid \mathbf{z}, \mathbf{s}, \mathbf{q})}_{\text{multiple choice}} \underbrace{P(\mathbf{z} \mid \mathbf{s}, \mathbf{q})}_{\text{open response}}, \quad (5.4)$$

where we adopt vector notation for the variables and parameters in our model to facilitate the development of the learning algorithm in Section 5.4. In order to give the dimensions for each of the variables in (5.4), assume that each student in  $S_{\text{open}}$  provides an open-response answer to each of the questions in  $Q$  and that each student in  $S_{\text{mcq}}$  also answers each question in  $Q$  (which entails providing a response to each option contained in a given question). Under these assumptions then,  $\mathbf{z} \in \{+1, -1\}^{|S_{\text{open}}||Q|}$ ,  $\mathbf{y} \in \{+1, -1\}^{|S_{\text{open}}||S_{\text{mcq}}||Q|}$ ,  $\mathbf{s} \in \mathbb{R}^{|S_{\text{mcq}} \cup S_{\text{open}}|}$  and  $\mathbf{q} \in \mathbb{R}^{|Q|}$ .

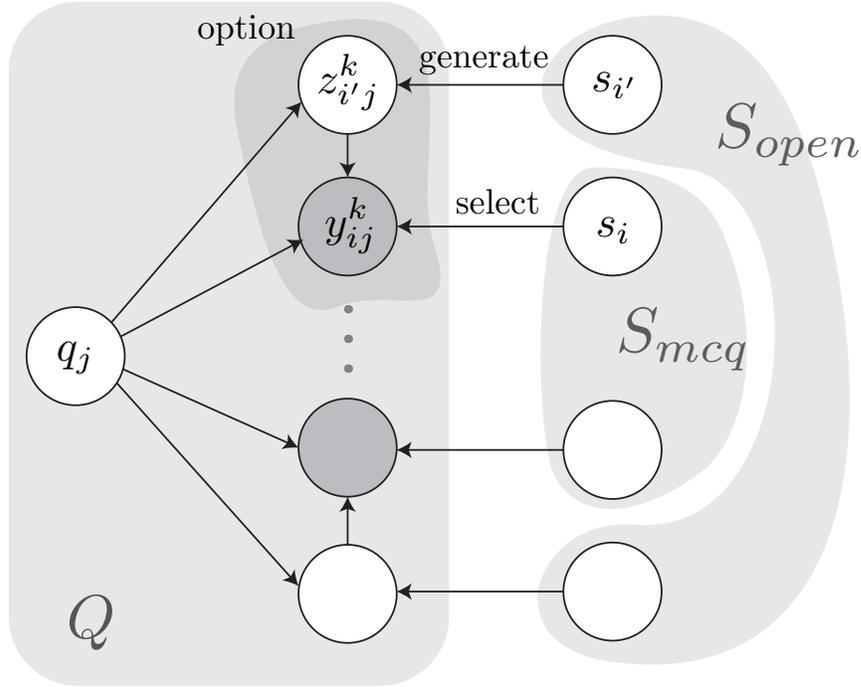


Figure 5.1: A probabilistic graphical model that summarizes the proposed *joint assessment and grading (JAG)* framework. Our model jointly captures the statistical dependencies between the abilities of students that generate open-response answers ( $S_{open}$ ), the abilities of students that select the correct answers when they are presented in a form of a multiple choice test ( $S_{mcq}$ ), the underlying question difficulty  $q_j$  and the observed responses ( $y_{ij}^k \in \{+1, -1\}$ ). The correctness of each open-response answer  $z_j^k \in \{+1, -1\}$  (omitting the index of student that generated the answer) is a hidden variable, the state of which is inferred in addition to the remaining parameters during inference. Note that the model is able to “grade” an open-response submission of a students by integrating the response patterns of other students who were presented that open-response answer as a choices in a multiple choice version of the question.

## 5.4 Parameter Learning

We now derive the expectation maximization (EM) algorithm for obtaining an approximate maximum likelihood estimate (MLE) of the parameters  $s$  and  $q$  of the model in (5.4). We briefly outline the key steps in obtaining the algorithm.

**E-step:** We compute the expectation of the log-likelihood (the logarithm of Equation 5.4) with respect to the unobserved variables  $\mathbf{z}$  which yields a function  $f(s, q)$  of the parameters  $s$  and  $q$  only. The expectation is performed with respect to the posterior distribution of  $\mathbf{z}$  given a previous estimate of  $s$  and  $q$  (or an initial guess).

**M-step:** We obtain an updated estimate of parameters  $s$  and  $q$  by maximizing  $f(s, q)$  obtained in the E-step.

The above procedure iterates until convergence. Below we give both steps explicitly in the context of the *joint assessment and grading (JAG)* framework.

**E-step:** Let  $\hat{s}$  and  $\hat{q}$  be an intermediate estimate of the parameters. Conditioning on these estimates, the posterior of  $z_j^k$  (correctness of answer (option)  $k$  to question  $q_j$ ) is a Bernoulli random variable with the probability of being correct given by (up to a normalizing constant):

$$P(z_j^k = 1 \mid \hat{s}, \hat{q}_j) \propto \underbrace{P(z_j^k = 1 \mid \hat{s}_i, \hat{q}_j)}_{\text{open response}} \prod_{s_i \in S_{\text{mcq}}} \underbrace{P(y_{i,j}^k \mid \hat{s}_i, \hat{q}_j, z_j^k = 1)}_{\text{multiple choice responses}}. \quad (5.5)$$

The posterior over the answer correctness  $z_j^k$  naturally integrates two sources of information: (i) the likelihood that the student who generated the answer was correct, and (ii) the likelihood that the students answering the multiple choice

version of the question “picked” this answer as correct (note that  $s_{i'} \in S_{open}$  and  $s_i \in S_{mcq}$ ). Each likelihood is parametrized by the model’s current estimate of the students’ abilities and question difficulties, and as a consequence gives more weight to the signal coming from the more able students.

**M-step:** The expectation of the log-likelihood with respect to  $\mathbf{z}$  yields the following expression:

$$\begin{aligned}
\mathbb{E}_{\mathbf{z}}[\log P(\mathbf{y}, \mathbf{z} \mid \mathbf{s}, \mathbf{q})] &= f(\mathbf{s}, \mathbf{q}) = \\
&= \sum_{D_{mcq}} P(z_j^k = +1 \mid \hat{\mathbf{s}}, \hat{q}_j) \underbrace{\log P(y_{i,j}^k \mid s_i, q_j, z_j^k = +1)}_{R_1} \\
&+ \sum_{D_{mcq}} P(z_j^k = -1 \mid \hat{\mathbf{s}}, \hat{q}_j) \underbrace{\log P(y_{i,j}^k \mid s_i, q_j, z_j^k = -1)}_{R_1} \\
&+ \sum_{D_{open}} P(z_j^k = +1 \mid \hat{\mathbf{s}}, \hat{q}_j) \underbrace{\log P(z_{i',j}^k = +1 \mid s_{i'}, q_j)}_{R_2} \\
&+ \sum_{D_{open}} P(z_j^k = -1 \mid \hat{\mathbf{s}}, \hat{q}_j) \underbrace{\log P(z_{i',j}^k = -1 \mid s_{i'}, q_j)}_{R_2}
\end{aligned}$$

where we introduce the short-hand  $D_{open}$  and  $D_{mcq}$  to refer to the sets of students, questions and responses that were involved in (i) generating open-response submissions and (ii) multiple-choice responses respectively. The above expression is a weighted linear combination of (log-) Rasch-likelihoods ( $R_1$  and  $R_2$  are given in (5.3) and (5.1) respectively), and can be easily maximized with a small modification to an existing Rasch solver to account for the constants. We use the L-BFGS algorithm [212] to perform this optimization step.

**Initialization:** Note that while the M-step is convex, the joint optimization problem in  $\mathbf{z}$ ,  $\mathbf{s}$ , and  $\mathbf{q}$  is not convex, and in general the EM algorithm will only yield an approximate solution and may get trapped in local optima. The problem becomes more pronounced in datasets with few interactions, e.g., small

classrooms. As such, initialization plays an important role in determining the quality of the obtained solution. A natural heuristic for initializing the posteriors over  $\mathbf{z}$  is with the fraction of “votes” given to the answer (i.e., fraction of students that identified the answer as correct). This heuristic is also suggested in [42]. We demonstrate the effectiveness of this heuristic in Section 5.6.

## 5.5 Experiments with Synthetic Data

In order to understand the behavior of our framework in a hypothetical classroom, we evaluate the model on a series of synthetically generated datasets. As our model attempts to infer the correctness of each answer entirely from the choices made by students in answering multiple choice questions, an important concern is the limitation of inference on difficult questions. Difficult questions are questions where we can expect the majority of students to be unable to identify the correct answers, and present a challenge to any model that relies on aggregating judgements. The model’s ability to recover the correct answer despite the majority being incorrect, fundamentally requires the model to leverage its estimates of students’ abilities so as to weigh the judgements of better students proportionally higher. Also note that we are concerned with questions of great *relative difficulty* (with respect to the ability of the students in the class), not absolute difficulty.

We can simulate an entire spectrum of regimes that present a varying degree of difficulty to inference, and evaluate the model’s performance in correctly inferring the correct answers in each regime. We accomplish this by generating a synthetic population of students and questions with a fixed expected *relative*

competency (i.e.,  $\mathbb{E}[s - q] = k$ , where  $s \sim p(s)$  and  $q \sim p(q)$ ), performing inference with our model on the generated observations, and computing the fraction of correctly inferred correct answers (accuracy) for different  $\mathbb{E}[s - q]$ . See Figure 5.2 for an illustration. Note that  $\mathbb{E}[s - q]$  is a quantity that conveniently summarizes the classroom in terms of its “competency” relative to the testing material. Large values of  $\mathbb{E}[s - q]$  indicate that the students are well-prepared, and most will answer the questions correctly.

### 5.5.1 Simulation procedure

We let  $p(s) = \mathcal{N}(\mu_s, \sigma = 2)$  and  $p(q) = \mathcal{N}(\mu_q, \sigma = 2)$ . We generate a synthetic classroom with the following parameters  $|S_{\text{open}}| = 10$ ,  $|S_{\text{mcq}}| = 10$ ,  $|Q| = 15$ , where every student in  $S_{\text{open}}$  submits an open-response answer to every question in  $Q$ , every student in  $S_{\text{mcq}}$  responds to every question (which entails providing a response to every option) and  $|S_{\text{mcq}} \cap S_{\text{open}}| = \emptyset$ . We then sample hidden ( $\mathbf{z}$ ) and observed ( $\mathbf{y}$ ) variables from Bernoulli distributions parametrized by (5.2) and (5.3) respectively.

Figure 5.3 illustrates the performance of the model as a function of the expected relative competency of the students ( $\mathbb{E}[s - q]$ ). We compare the performance of our model to a simple *majority* baseline (i.e., label the answer as correct if the majority of the students select it). As expected, the majority baseline works best when the relative competency of the class is high (since most students will correctly identify the correct answers). The performance degrades significantly in the regime where the relative competency is negative (i.e., most students are expected to answer the questions incorrectly). Observe that the model is able to

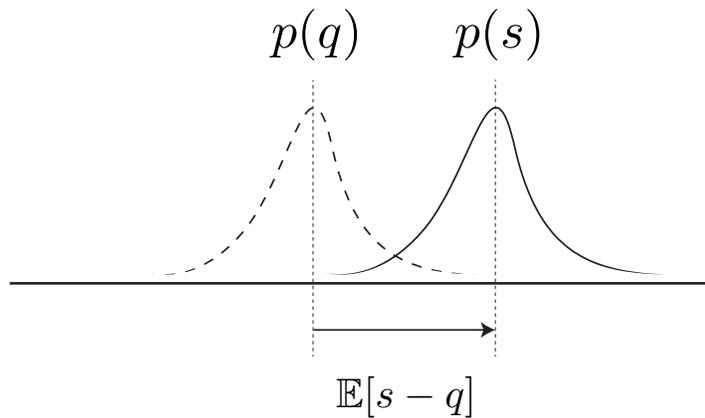


Figure 5.2: Distributions used in generating synthetic data, where  $p(q)$  and  $p(s)$  are the distributions of question difficulty and student ability respectively. The quantity  $\mathbb{E}[s - q]$  represents the average relative competency of the classroom: a large value of  $\mathbb{E}[s - q]$  indicates that the majority of students will answer most of the test items correctly, and vice-versa. See Figure 5.3 for the effect of the class distribution on performance.

maintain a significant performance margin ( $>10\%$ ) over the baseline even in the regime of low relative competency.

## 5.6 Real-World Experiments

We emulate a classroom setting on the Amazon Mechanical Turk platform by soliciting Mechanical Turk workers to participate in a reading comprehension task. The study was conducted in two separate phases with a different set of workers in each: (i) the *open-response task* and (ii) the *multiple choice task*. In each task, a worker was presented with an article<sup>1</sup>, followed by a set of 15 questions. In the *open-response task*, the questions were displayed in an open-response format,

<sup>1</sup>Unit 7.2 (Language) from the OpenStax Psychology textbook

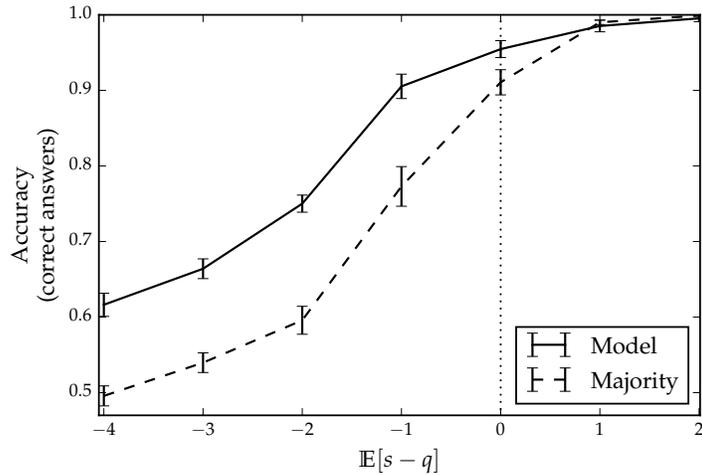
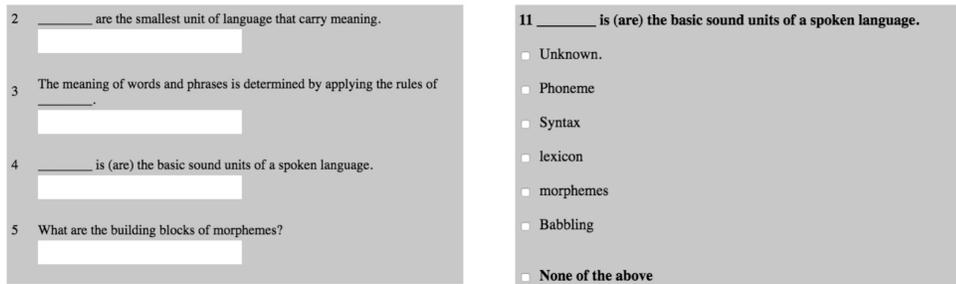


Figure 5.3: Accuracy in predicting the correct answers on synthetic data, as a function of the average relative competency in the classroom (measured in the multiples of standard deviations of the distributions). The simple majority-vote baseline performs comparably with our model for class distributions with large relative competency (since the majority of the students answer most questions correctly). The model significantly outperforms the baseline in the regime of lower relative competency (i.e., when most questions are too difficult for the majority of the students).

and the workers were asked to type in their response. In the *multiple choice task*, the same 15 questions were presented in a multiple-choice format, with the choices aggregated from the open-response submissions obtained in the *open response task*. The answers collected in the *open response task* were clustered semi-automatically before being displayed as choices in the *multiple choice task*. The clustering step aggregated identical answers or answers within a few characters in difference (for example due to spelling errors), and semantically identical answers were then grouped manually (e.g., paraphrases). Clustering answers is a critical pre-processing step as it ensures that a reasonable number of choices is shown as part of the multiple choice question, as well as that the conditional



(a) *open response task*

(b) *multiple choice task*

Figure 5.4: Screenshot of a segment from each of the two Mechanical Turk tasks. Workers are required to provide an open response answer to each question in the *open response task*, and select (click) all answers that apply in the *multiple choice task*. The choices in the *multiple choice task* are aggregated from the open-response submission of other workers as part of the *open-response task*.

independence assumption discussed in Section 5.3 holds. In the next chapter, we will introduce an extension to the model for automatically clustering open-response submissions based on the multiple choice response signal alone.

In total, 15 workers participated in the *open-response task* and 82 workers participated in the *multiple choice task*. A total of 225 open-response submissions were generated in response to the total of 15 comprehension questions, resulting in 101 distinct choices after clustering.

### 5.6.1 Results

We evaluate the effectiveness of our algorithm on the data collected via Amazon’s Mechanical Turk using two performance metrics: (i) accuracy in predicting the correctness of each answer and (ii) quality of the predicted ranking of the

students. We evaluate our algorithm in a *semi-supervised* setting where we provide a set of partially labeled items, i.e., we label correctness for a subset of the answers. This represents a practical use-case of our framework—instead of being entirely hands-off, an instructor may choose to manually grade a subset of the students’ answers to improve the performance of automatic inference. We evaluate two versions of our model: **EM +open** and **EM -open** in addition to the majority baseline described in Section 5.5:

- **EM +open**: The full model as described in Section 5.3 and Section 5.4.
- **EM -open**: A subset of the **EM +open** model lacking the *open-response* component described in Section 5.3.2. In other words, during inference the model does not leverage any information about the ability of the answer generator, and relies entirely on the multiple choice responses to infer the correctness of the answers.

### Predicting answer correctness

Figure 5.5 depicts accuracy as function of the amount of labeled data (accuracy was computed with respect to a gold-standard annotation of correctness for each answer, performed by the author). From it we conclude that (i) the full model (**EM +open**) significantly outperforms both the majority baseline and **EM -open**, (ii) the **EM +open** performs very well without any labeled data ( $\approx 86\%$  accuracy), (iii) adding labeled data improves performance, and (iv) the *open-response* component of the model (one that is lacking in the **EM -open** model) is critical in significantly boosting performance, i.e., incorporating information about the answer creator is valuable in inferring the correctness of each answer.

**Initialization:** We also note that the initialization heuristic suggested in Section 5.4 is critical to achieving competitive performance in the regime of little to no labeled data (Figure 5.6). The performance of the model drops significantly below the majority baseline when a random initialization is used in place of the suggested heuristic.

### Predicting student ranking

Although predicting the correctness of each answer is itself a valuable intermediate output, a motivating use-case of our framework is to assess the students' competency. A ranking of the students by their expertise is one example of summative assessment, and may be valuable in identifying students that excel or are in need of additional help. We evaluate the quality of the rankings produced by our model in the following way: (i) use the gold-standard annotation for the correctness of each answer to fit a standard Rasch model, identifying the abilities  $s_{\text{gold}}$  of each student (both in  $S_{\text{open}}$  and  $S_{\text{mcq}}$ ), (ii) obtain the ability parameters using our model (**EM +open** and **EM -open**) (trained with a varying amount of labeled data) and (iii) rank the students according to each set of parameters and compute rank correlation. We use Kendall-Tau as a metric of rank correlation. Kendall Tau returns a quantity in the range  $[-1, +1]$ , where  $+1$  indicates perfect correlation (every pair of students in both rankings are in a consistent order),  $-1$  when the rankings are inverted, and  $0$  when the rankings are not correlated.

Figure 5.7 and Figure 5.8 depict rank correlation as a function of the amount of labeled answers for the students in sets  $S_{\text{open}}$  (workers in the **open response task**) and  $S_{\text{mcq}}$  (workers in the **multiple choice task**) respectively. We observe that (i) incorporating partially labeled set of answers improves rank correlation, (ii) the

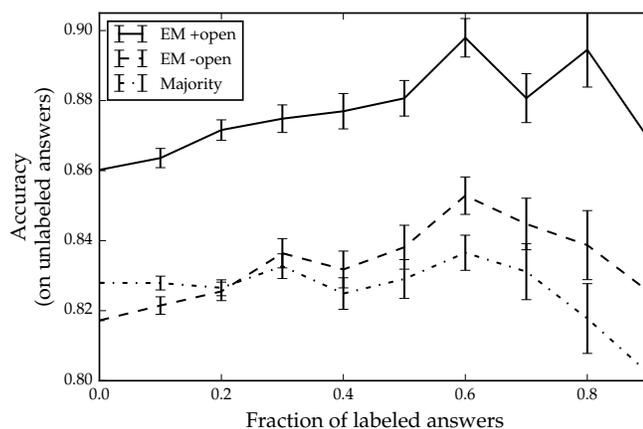


Figure 5.5: Accuracy in predicting the correct answers in the dataset collected on Mechanical Turk. The model that incorporates both the *open-response* and *multiple choice* components (**EM +open**) significantly outperforms the model that only incorporates the multiple choice component (**EM -open**) and a simple majority-vote baseline.

**EM +open** model performs superior to or on par with the majority baseline (note that **EM -open** is not relevant when ranking the students in the  $S_{open}$  set).

### Effect of classroom size

In a practical setting, it is important to consider the effect of classroom size on the quality of the inferred parameters. Intuitively, we expect that increasing the number of students answering multiple choice questions  $|S_{mcq}|$  will improve performance (accuracy and rank correlation). Figure 5.9 depicts accuracy as a function of  $|S_{mcq}|$  (number of students that answer multiple choice questions), for two conditions based on the amount of partially-labeled answers available. As expected, we observe that the performance of the model (**EM +open**) increases when more students participate in answering MCQs, and the gain increasing with the number of labeled answers.

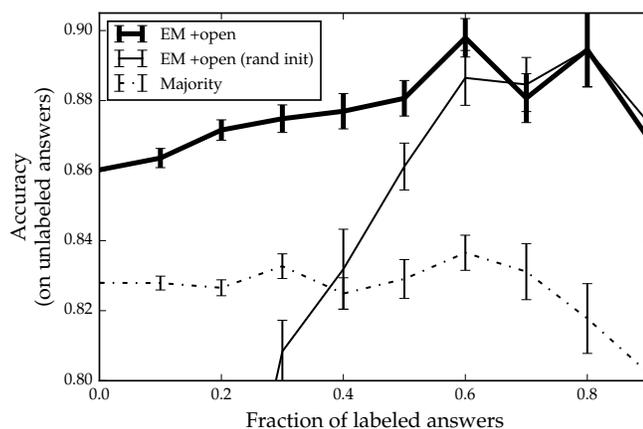


Figure 5.6: Accuracy in predicting the correct answers in the dataset collected on Mechanical Turk for the model initialized with the heuristic described in Section 5.4 (**EM +open**) and the model initialized randomly (**EM +open (rand init)**). Good initialization significantly improves performance, especially in the regime of little to no partially labeled data.

## 5.7 Conclusion

In this chapter, we have developed a novel framework for crowdsourced content generation and evaluation. In the context of education, our framework offers a powerful alternative to classical peer-grading, as it naturally fuses test-taking and grading into a unified, streamlined process with a common incentive mechanism. Furthermore, our framework is general enough to be applied in different crowdsourcing tasks where the goal is to generate and identify quality contributions. Our framework also opens the door to a natural way of automatically clustering solutions — a task that forms the focus of the next chapter.

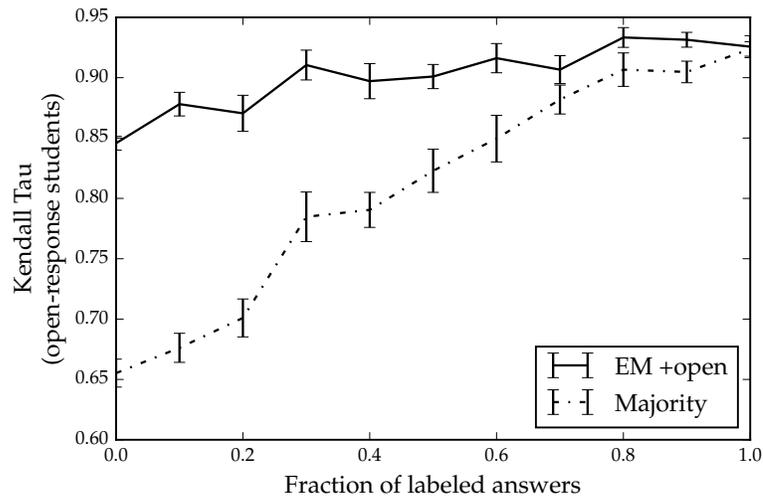


Figure 5.7: Rank correlation (kendall-tau) for students submitting open-response answers ( $S_{open}$ ) between the model-inferred ranking (**EM + open**) and the ranking obtained using the gold-standard correctness labels for each answer (via the Rasch model). The model generates high quality rankings with little to no labeled data, significantly outperforming the majority baseline where students are ranked using the parameters obtained from the Rasch model, but where the correctness of each answer is obtained via a majority vote).

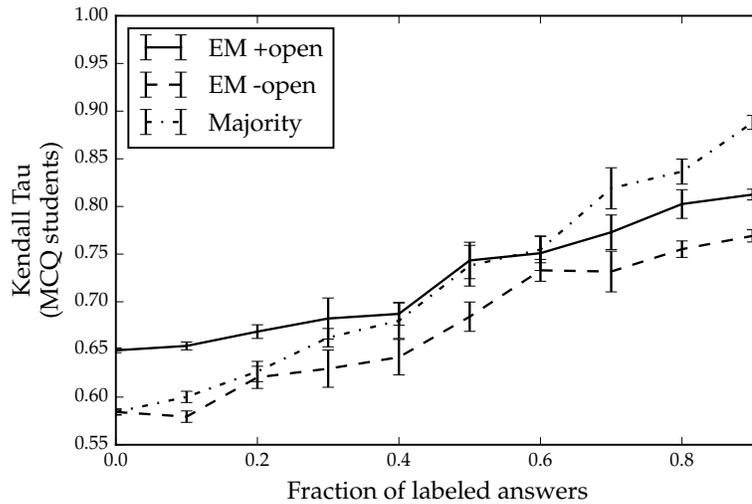


Figure 5.8: Rank correlation (kendall-tau) for students submitting multiple choice answers ( $S_{mcq}$ ) between the model-inferred ranking (**EM +open** and **EM -open**) and the ranking obtained using the gold-standard labels for each answer (via the Rasch model). In contrast to the rank correlation for students submitting open-response answers (Figure 5.7), rank correlation for multiple choice students is lower.

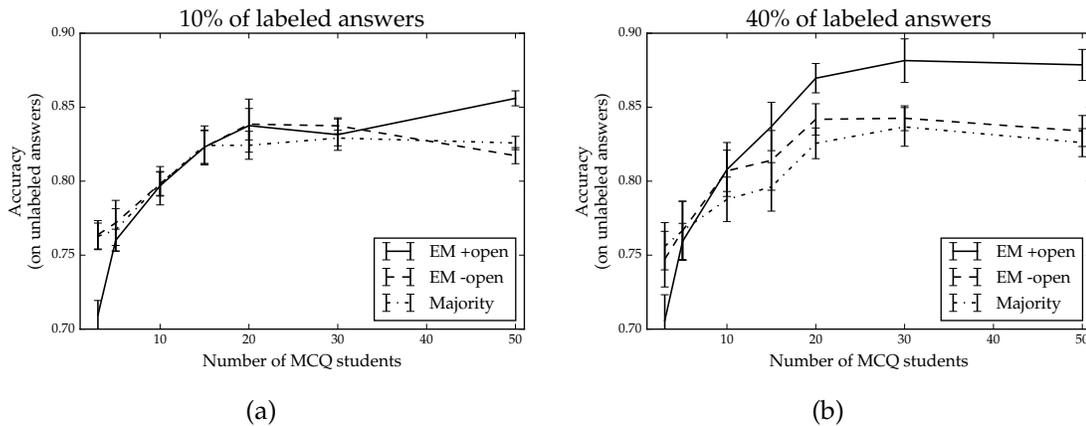


Figure 5.9: Accuracy in predicting the correct answers in the dataset collected on Mechanical Turk as a function of the number of students answering multiple choice questions ( $|S_{mcq}|$ ). More students answering multiple choice questions improves the performance of the model (**EM +open**) in relation to the majority baseline.

## CHAPTER 6

# LATENT CHOICE ALLOCATION: SOLUTION CLUSTERING FROM IMPLICIT FEEDBACK

### 6.1 Introduction

In this chapter, we address the problem of clustering open-response submissions — a key limitation of the Joint assessment and grading (JAG) framework described in the previous chapter. We propose a novel domain-independent framework, referred to as latent choice allocation (LCA), for automatically clustering and grading open-ended submissions in aptitude tests based only on the students’ responses to multiple choice questions constructed from open-response submissions of other students. Our model relies only on the selections made by the students attempting to answer the MCQs in order to (i) automatically cluster similar (or equivalent) open-response answers and (ii) grade them, i.e., identify the clusters of answers that are correct and incorrect. We evaluate our framework using simulations and demonstrate its efficacy using real-world experiments carried out on Amazon Mechanical Turk.

One of the long-standing challenges towards scaling education beyond the “physical classroom” is the problem of efficiently assessing a large number of students. Conventional aptitude tests, typically designed and graded by course instructors, have poor scaling properties in classrooms with disproportionate student-teacher ratios—this is particularly true in extreme settings such as massive open online classrooms (MOOCs) with hundreds to thousands of students. As a result, the problem of *scalable assessment* has gained significant attention in the fields of data mining, machine learning and natural language process-

ing [143, 148, 9, 24, 103, 86].

For MOOCs two well-established approaches are available that enable large-scale assessment: *multiple choice testing* and *peer grading*, each with its own advantages and disadvantages. In this chapter, we extend the alternative approach to scalable assessment proposed in the previous chapter (JAG) to also automatically cluster students' submission. In order to lay the foundation of our framework, we first briefly outline the limits of the existing methods.

### **6.1.1 Multiple choice testing**

Multiple choice questions (MCQs) circumvent the scaling problem all-together since they do not require any manual grading effort; this explains their wide adoption in MOOCs. This very feature that makes MCQs attractive for grading large classroom settings also makes them notoriously difficult to design in practice [67]. Constructing a set of viable alternative options (distractors) that probe likely misconceptions among students requires sufficient experience and effort on the part of the instructor. In fact, MCQ writing is known to be both an art and science, with a plethora of instructional guidelines available [70, 67, 69]; poorly designed MCQs can be at best uninformative, and at worst misleading in identifying the gaps and misconceptions in students' learning.

A more fundamental issue, however, is that even well-designed MCQs are incapable of facilitating the kind of higher-order thinking and student engagement characteristic to open-ended (or open-response) questions [67], where the student is required to come up with their own answer. Thus, developing scalable methods to administer open-ended questions in MOOCs is critical for improving

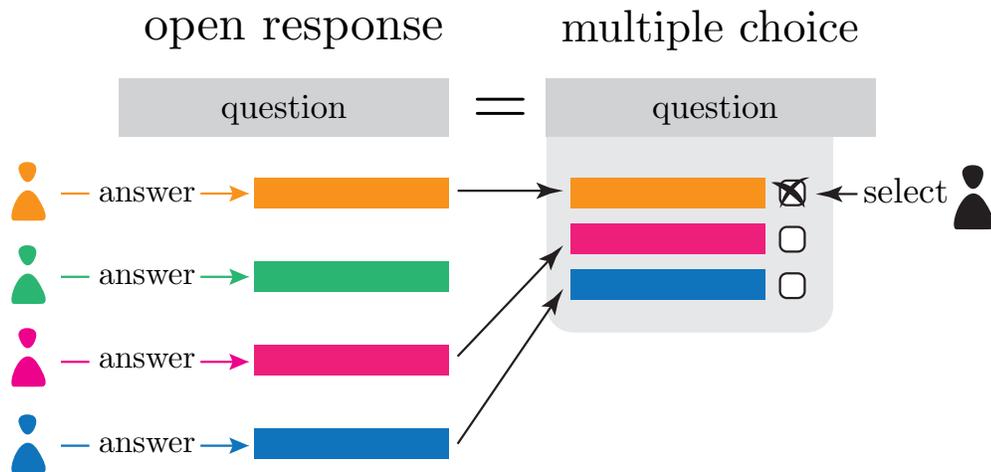


Figure 6.1: A high-level overview of the proposed latent choice allocation (LCA) framework: open-response submissions are combined into a multiple choice question; selections made by those students answering a multiple choice version of the question are used as “feedback” to (i) cluster similar open-response submissions and (ii) grade them, i.e., identify correct and incorrect open-response submissions. From the perspective of the students, both groups are taking a test (either open-response or multiple choice), but those answering the multiple choice questions are also implicitly grading and clustering the open-response submissions.

the learning experience.

### 6.1.2 Peer-grading

Peer-grading addresses the problem of scalable grading of open-response submissions by delegating the role of a grader to every student in the class. The key technical challenge of peer-grading—one that has been receiving the most attention from the machine learning and data mining communities—is that of *score aggregation*, i.e., deploying an optimal strategy to combine scores assigned

by different students with widely varying “grading abilities” into a final grade.

Traditional peer-grading, where each student in the class grades a subset of their peers’ assignments, suffers from several drawbacks. First, in addition to doing their own assignments, each student now faces an additional burden of grading other students’ work. While the effort spent on their own assignments is directly incentivized by a grade, devising an incentive mechanism for eliciting effort in grading other students’ work is difficult. For example: should the students be rewarded with points for grading other assignments, and should that reward depend on how “well” they grade? If so, how does one define “well” in the context of grading? Second, although traditional peer-grading is far more scalable than instructor-only-grading, it faces a theoretical ceiling on the accuracy of the aggregated scores. As shown in [174], regardless of the size of the classroom, conventional peer-grading will always result in a fraction of misgraded assignments. A natural solution to more scalable peer-grading (i.e., with no misgraded assignments), as shown theoretically in [174] and demonstrated in practice by [24, 103] is *clustering*: grouping similar assignments of multiple students and grading the entire group simultaneously. Clustering submissions in a domain-independent way, however, is a difficult task. The successful realizations of clustering exploit domain-specific structure (natural language and mathematical expressions, respectively), where clustering is performed based on the features that describe the submissions and the structure of the domain (e.g., words and syntax, mathematical operators, and rules of algebra). Extending these methods to other domains, e.g., clustering circuit diagrams, chemical formulas, or computer code, requires dedicated, domain-specific solutions.

### 6.1.3 Proposed framework and contributions

Motivated by traditional multiple choice testing and peer-grading, we propose to combine *assessment and grading*, which enables us to exploit the advantages of both approaches. Our key idea is that grading and clustering can be made implicit when the submissions of other students are disguised as a multiple choice question for another student, where the options of the MCQ are precisely the open-response answers to the same question (and where multiple options are allowed to be correct). As a result, a student answering a multiple choice question constructed from the responses of other students is implicitly (i) “grading” these open-response submissions (by selecting the responses he or she thinks are correct) and (ii) clustering responses that are similar (by selecting multiple responses together). In a large classroom, we expect that many open-response submissions will be redundant, e.g., paraphrases, equivalent mathematical expressions, functionally-equivalent circuit diagrams, etc. When a student is presented with such redundant options in the form of a multiple choice question, they will invariably select (i.e., identify as correct) all of such semantic variations together. Thus, the pattern of multiple choice options that tend to be always selected together is a “signal” that identifies which underlying open-response answers are such semantic variations (and should be clustered together). In effect, our model leverages the students themselves to “do the clustering,” rather than relying on domain-specific machine learning techniques. This feature makes our approach domain-independent and potentially more robust. Finally, a powerful side-effect of the proposed mechanism is that questions will—by design—consist of common and relevant misconceptions in the class, a fundamental guiding principle of MCQ design [69].

Our main contributions can be summarized as follows:

- We develop a non-parametric Bayesian framework, referred to as latent choice allocation (LCA), for automatically identifying groups (clusters) of *latent answers* among students' open-response submissions. Our framework only relies on the observation of the choices made by other students when the open-response submissions are shown in the form of a MCQ. To the best of our knowledge, this is the first demonstration of domain-independent clustering applied to open-ended learning content.
- We evaluate our model in a real-world “classroom” on Amazon Mechanical Turk, and demonstrate that LCA is effective at (i) finding groups of semantically similar answers and (ii) identifying groups of correct and incorrect answers (i.e., grading open-response submissions).

## 6.2 LCA: Latent Choice Allocation

In this section, we describe the probabilistic LCA framework for automatically clustering and grading open-response submissions. To simplify the exposition of our ideas, we first introduce a simpler setting that excludes latent answers and clustering, i.e., where every open-response submission can be considered sufficiently distinct from all others. We then propose the full-fledged LCA model that includes clustering.

## 6.2.1 Model without latent answers

Each student  $i \in S$  is endowed with a real-valued ability parameter  $s_i \in \mathbb{R}$ , such that students with greater ability are more likely to identify the correct answers in a set of existing answers (submissions). Each answer (submission)  $c \in C_j$  to question  $j \in Q$  is endowed with answer-specific parameters  $\{\beta_{jc}\}$ , which can be interpreted as the “difficulty” of answer (submission)  $c$  to question  $j$ . The “difficulty” of an answer determines the likelihood of a student with a fixed ability to correctly identify this particular answer as either correct or incorrect. This model accounts for the fact that questions might have multiple correct answers, but some could be easy to identify as correct or incorrect, and others might be tricky distractors that require better knowledge of the material. We define the likelihood of student  $i$  to select a particular answer  $c$  of question  $j$  as correct choice as follows:

$$P(x_{ijc} \mid s_i, \beta_{jc}, y_{jc}) = \frac{1}{1 + \exp(-x_{ijc}y_{jc}(s_i - \beta_{jc}))}, \quad (6.1)$$

where  $x_{ijc} \in \{-1, +1\}$  indicates whether student  $i$  selected ( $x_{ijc} = +1$ ) choice  $c$  in question  $j$  and  $y_{jc} \in \{-1, +1\}$  is a variable that identifies whether answer  $c$  in question  $j$  is a correct answer ( $y_{jc} = +1$ ). Intuitively, if the student’s relative ability (difference between their absolute ability  $s_i$  and answer difficulty  $\beta_{jc}$ ) is large, *and* the answer is a correct answer ( $y_{jc} = +1$ ), the student will have a high probability of selecting that answer (i.e.,  $x_{ijc} = +1$ ). By symmetry, a student with a large relative ability will have a high probability of *not* selecting the answer if the answer is wrong, i.e.,  $y_{jc} = -1$ . In the psychometrics literature, a likelihood of this form is known as the Rasch model [149]. This model is, however, commonly used for responses at a question-level (rather than at a choice-level) and the correctness of each response ( $y_{jc}$ ) is assumed to be known. In what follows, we

will use terms *submission*, *answer*, and *choice* interchangeably depending on the context.

## 6.2.2 Model with latent answers

To motivate the necessity of extending the above model with the notion of *latent answers*, consider the following consequence of modeling the complete set of responses of student  $i$  to each choice  $c \in C_j$  of question  $j$  as:

$$P(\{x_{ijc}\}_{c \in C_j} \mid s_i, \{\beta_{jc}\}, \{y_{jc}\}) = \prod_{c \in C_j} P(x_{ijc} \mid s_i, \beta_{jc}, y_{jc}). \quad (6.2)$$

Here,  $P(x_{ijc} \mid s_i, \beta_{jc}, y_{jc})$  is given in (6.1) and we have made an assumption of conditional independence between the individual choices of the same student—conditioned on the student’s ability. While the assumption of conditional independence is justified in the case where every answer is distinct, it is violated when some of the answers are identical or otherwise semantically similar. For example, equivalent choices are “was born in St. Louis” and “birthplace is St. Louis,” as well as  $y = 2x^2 + z$  and  $y = z + 2x^2$ . The unintended consequence of assuming conditional independence in this case, is either over- or under-estimation of the student’s ability (with the severity depending on the number of such “redundant” answers). For example, if the same student selects three correct answers which are identical, a model that assumes conditional independence will effectively (and incorrectly) “perceive” this student answering three different questions correctly (and in effect inflate the model’s estimate of that student’s ability).

A more natural interpretation of how a real student might behave when faced with a set of such “redundant” answers (answers that are semantically identical)

is as follows: a student observes a set of choices in a multiple choice question and recognizes those answers that are semantically identical. The student then decides whether the entire group (cluster) of such semantically identical answers is correct (or incorrect), and then either selects (or does not select) *all* of the choices represented by that semantic group (in contrast to making a selection for each individual choice in the group). In effect, the student now reasons and makes decisions at the level of *latent answers*, rather than the observed answers, and then “copies” their decision to all of the observed answers grouped under that *latent answer*. We can capture this intuition by introducing a latent variable for each observed answer (i.e., each open-response submission) that indicates its membership to one of the possible *latent answers*.

More formally, we can model this intuition by instead attributing the answer correctness and answer difficulty parameters ( $y_{jc}$  and  $\beta_{jc}$  in (6.1) to the level of *latent answers* rather than the observed answers as done in (6.1)). In what follows, we will index latent answers with  $k$ , and thus latent answer parameters with  $y_{jk}$  and  $\beta_{jk}$  (correctness and difficulty of latent answer  $k$  in question  $j$  respectively). To relate each observed answer (i.e., open-response submission) to its “parent” latent answer, we introduce a set of membership latent variables  $\{z_{jc}\}$  for each choice  $c$ , where  $z_{jc} \in \mathbb{N}$  is an index of the cluster (latent answer) to which choice  $c$  of question  $j$  belongs.

We now seek to formally express the likelihood of observing the response of student  $i$  to every answer (choice)  $c$  of question  $j$ , accounting for the latent structure. But first we define some additional notation to aid in conveniently formalizing this likelihood. Conditioned on the latent membership assignment  $\{z_{jc}\}$  of each choice  $c$  of question  $j$ , let  $C_j^S$  be a set of subsets of  $C_j$  (recall that  $C_j$

is the set of answers, or choices, of question  $j$ ) that partitions elements (answers) of  $C_j$  according to their latent answer membership (given by  $\{z_{jc}\}$ ). Formally, we have  $C_j^S = \{C_j^{(k)}\}$  where  $C_j^{(k)} = \{c \mid c \in C_j \wedge z_{jc} = k\}$  and such that  $\bigcup_k C_j^{(k)} = C_j$ . Similarly, let  $X_{ij} = \{x_{ijc}\}$  be the set of observed responses of student  $i$  on each answer (choice)  $c$  of question  $j$ , where  $x_{ijc} \in \{-1, +1\}$ , i.e.,  $+1$  if the student selected the choice and  $-1$  otherwise. Finally, let  $X_{ij}^S = \{X_{ij}^{(k)}\}$  be a set of subsets of  $X_{ij}$  partitioned by the latent assignment of each choice  $c$ , i.e.,  $X_{ij}^{(k)} = \{x_{ijc} \mid x_{ijc} \in X_{ij} \wedge c \in C_j \wedge z_{jc} = k\}$  and  $\bigcup_k X_{ij}^{(k)} = X_{ij}$ . Armed with this notation, we can now formally specify the likelihood of observing the response of student  $i$  to every answer (choice) in question  $j$  as follows:

$$P(X_{ij} \mid s_i, \{\beta_{jk}\}, \{y_{jk}\}) = \prod_{X_{ij}^{(k)} \in X_{ij}^S} P(X_{ij}^{(k)} \mid s_i, \beta_{jk}, y_{jk}). \quad (6.3)$$

Here, in contrast to (6.2) where the assumption of conditional independence is made across every answer (choice), we have assumed conditional independence only across groups (i.e., latent answers or clusters), but *not* within groups. However, this now leaves us with the task of defining the likelihood of the choice observations  $\{x_{ijc}\}$  within groups, i.e.,  $P(X_{ij}^{(k)} \mid s_i, \beta_{jk}, y_{jk}), \forall X_{ij}^{(k)} \in X_{ij}^S$ .

As stipulated at the beginning of this section, we assume that a student reasons and makes their decision about the correctness of each answer at the “latent level,” and then “copies” their decision to all of the answers (choices) grouped under that latent answer. Therefore, a natural likelihood function for  $P(X_{ij}^{(k)} \mid s_i, \beta_{jk}, y_{jk})$  would simply be the likelihood of the student’s response given the ability  $s_i$  of the student, and the difficulty  $\beta_{jk}$  and correctness  $y_{jk}$  of the latent answer, regardless of the number of answers (choices) represented by the latent answer. Effectively, this would be the likelihood given in (6.1), with

the choice-level parameters  $\beta_{jc}$  and  $y_{jc}$  replaced with the latent-level parameters  $\beta_{jk}$  and  $y_{jk}$ . We would, however, quickly run into a problem in attempting to directly adapt the likelihood in (6.1) to the latent level. The likelihood given in (6.1) would require us to assume a single “representative” response for any given latent answer  $k$ , i.e., one that we assumed the student had first made up in their mind, and then “copied” to all of the observed answers grouped under that latent answer. In other words, this would require the students to always be “consistent” in submitting identical responses to all of the members of the same latent answer. A model that assumed that a priori, would invariably fail to recognize an underlying cluster among a subset of answers, even if only one student responded “inconsistently” (i.e., submitted different responses for a set of answers belonging to that latent answer). A natural solution to overcome this issue is to introduce “noise” into the likelihood function by assuming that each student has a potential to “fail” in recognizing the semantic similarity among a subset of answers (choices) that actually belong to the same latent answer. This is not necessarily an artificial construct to aid in model inference, but has a natural justification: different students may have a different concept of semantic similarity, e.g., one student may consider the answers sufficiently different when the wording is changed in a subtle way, while another student may not consider the particular change in wording to affect the meaning of the answers.

Formally, we can introduce “noise” into the likelihood function as follows: consider that for a subset of answers  $C_j^{(k)}$  grouped under the same latent answer  $k$ , a student  $i$  with probability  $\epsilon$  will fail to recognize that the choices are actually semantically identical. In that case, that student will respond to each answer individually, i.e., as if each answer was distinct. In the other case (with probability  $1 - \epsilon$ ), the student (i) *will* recognize the presence of a latent answer, (ii) make up *one*

response in their mind, and (iii) “copy” that response to every choice grouped under that latent answer. Formally, let  $t_{ijk} \sim \text{Bernoulli}(1 - \epsilon)$ , i.e., a hidden variable that indicates whether the student  $i$  will ( $t_{ijk} = 1$ ) recognize the latent answer  $k$  in question  $j$ . We can then express the likelihood of observing a set of responses to all choices grouped under latent answer  $k$ ,  $P(X_{ij}^{(k)} \mid s_i, \beta_{jk}, y_{jk}, t_{ijk})$  in (6.3) as follows:

$$P(X_{ij}^{(k)} \mid s_i, \beta_{jk}, y_{jk}, t_{ijk}) = \begin{cases} P(x_{ijc} \mid s_i, \beta_{jk}, y_{jk}) & \text{if } t_{ijk} = 1 \\ \prod_{x_{ijc} \in X_{ij}^{(k)}} P(x_{ijc} \mid s_i, \beta_{jk}, y_{jk}) & \text{otherwise} \end{cases} \quad (6.4)$$

where  $P(x_{ijc} \mid s_i, \beta_{jk}, y_{jk})$  is given by (6.1), with the exception that now  $\beta_{jk}$  and  $y_{jk}$  are *latent answer* “difficulty” and correctness rather than *observed* answer “difficulty” and correctness as in (6.1).

### 6.2.3 Dirichlet process (DP) prior for latent answers

The Dirichlet process (DP) prior is a flexible prior for non-parametric mixture models, i.e., mixture models where the number of clusters (in our case latent answers) is not specified a priori, and can grow with the data [134]. A non-parametric mixture model is particularly suitable for modeling latent answers for two reasons: (i) we expect the number of truly distinct (original) open-response answers for any given question to be smaller than the number of contributors (students), but we do expect it to grow as more students contribute (this is naturally modelled with the DP) and (ii) each question can be expected to have a different number of latent answers (clusters); in this case an alternative to the DP, such as model-selection in a finite mixture model, becomes intractable, as

the search space becomes exponential in the number of questions.

For the purpose of modelling and inference in our setting, it is most convenient to work with the conditional distribution over latent answers with the DP prior marginalized out, i.e.,  $P(z_{jc} | \mathbf{z}_j^{-c}, \mathbf{x}_j, \mathbf{s}, \phi_j)$ , known as the Chinese restaurant process (CRP) in literature [134]. We use  $\phi_j$  to represent cluster-specific parameters for question  $j$ , which can be written explicitly as:  $\{(\beta_{jk}, y_{jk})\}_{k=1, \dots, \infty}$ , i.e., “difficulty”  $\beta_{jk}$  and correctness  $y_{jk}$  for each latent-answer (cluster). We use the vectors  $\mathbf{x}_j$  to encode the response of every student to every answer (choice) of question  $j$ , i.e.,  $\{x_{ijc}\}_j$ ,  $\mathbf{z}_j$  to encode the latent answer membership of every observed answer (choice) of question  $j$ , i.e.,  $\{z_{jc}\}_j$  and  $\mathbf{z}_j^{-c}$  to represent the latent membership of every observed answer, except  $c$ . Finally,  $\mathbf{s}$  is a vector encoding the student ability parameters, i.e.,  $\{s_i\}$ . The CRP gives a conditional distribution over the latent variable membership of choice  $c$  of question  $j$  as:

$$\begin{aligned}
 P(z_{jc} | \mathbf{z}_j^{-c}, \mathbf{x}_j, \mathbf{s}, \phi_j) &\propto & (6.5) \\
 \sum_k^K \frac{n_{jk}}{\alpha + n_j - 1} P(\mathbf{x}_j | \mathbf{z}_j^{(z_{jc}=k)}, \mathbf{s}, \phi_j) \delta(z_{jc} = k) &+ \\
 \frac{\alpha}{\alpha + n_j - 1} \left( \int_{\phi} P(\mathbf{x}_j | \mathbf{z}_j^{(z_{jc}=k')}, \mathbf{s}, \phi) G_0 d\phi \right) &\delta(z_{jc} = k').
 \end{aligned}$$

Here,  $n_{jk}$  is the number of observed answers to question  $j$  assigned to latent answer  $k$ ,  $n_j$  is the total number of answers of question  $j$ , and  $\alpha$  is the hyperparameter of the DP. Furthermore, the notation  $\mathbf{z}_j^{(z_{jc}=k)}$  represents the latent assignment of answers of question  $j$ , with choice  $c$  assigned to latent answer  $k$ . We use  $k'$  to refer to the index of a latent answer which does not yet exist (i.e., new cluster), and  $K$  is the total number of clusters that have had at least one choice assigned to it. Finally,  $G_0$  is the base distribution of the DP, i.e., a prior over the parameters of the latent answers. Because in our model, each latent answer is endowed with two parameters, difficulty  $\beta_{jk}$  and correctness  $y_{jk}$ ,

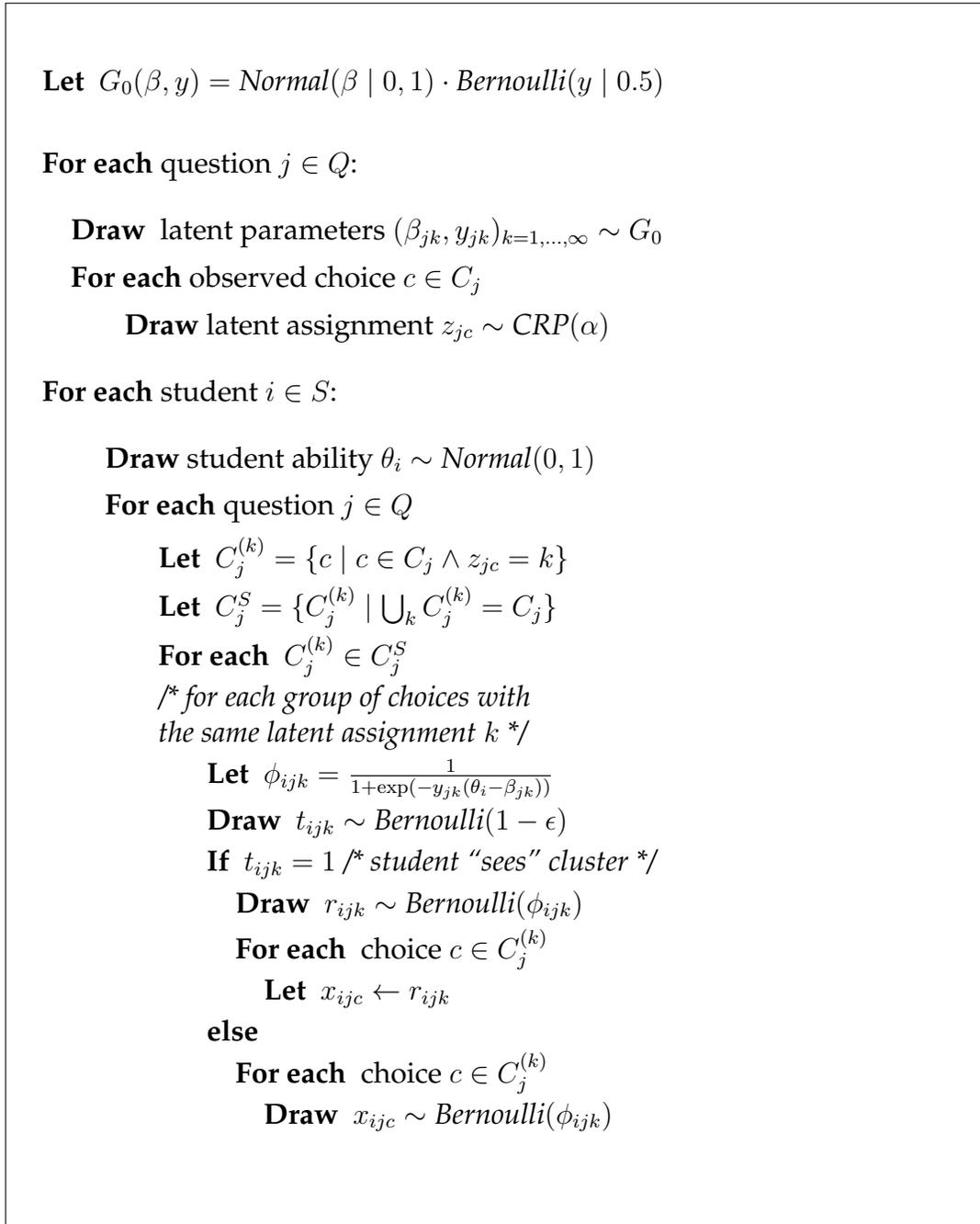


Figure 6.2: Generative process of the proposed model.

$G_0$  is a joint distribution over  $(\beta_{jk}, y_{jk})$ , where for convenience, we assume that  $\beta_{jk} \sim Normal(0, 1)$  and  $y_{jk} \sim Bernoulli(0.5)$ . The complete generative process is summarized in Figure 6.2.

## 6.2.4 Parameter inference

We use a combination of Metropolis-Hastings (MH) and Gibbs sampling to perform parameter inference in this model. MH is used for the continuous variables  $\{s_i\}$  and  $\{\beta_{jk}\}$  (as our likelihood does not possess a conjugate prior), and Gibbs sampling is used for  $\{y_{jk}\}$  and  $\{z_{jc}\}$ . The integral in (6.5) is computed numerically using a conventional quadrature method.

## 6.3 Synthetic Experiments

We first perform two simulation studies to validate the correctness of our model.

### 6.3.1 Experiment 1: Extreme configurations

Consider an extreme latent answer configuration that would present the most challenging case for inference in our model: multiple latent answers (clusters) with identical parameters (i.e., difficulty and correctness). When presented with this configuration, the model cannot rely on the differences in the parameters (difficulty and correctness) of the latent answers to aid in disambiguating between the two competing hypotheses, for example (a) two clusters vs. (b) four clusters, but with two pairs of clusters having an identical set of parameters (see left-most panel of Figure 6.3 for an illustration). In this regime, the model can only leverage the differences in observations arising as a consequence of the different underlying latent structures among answers.

To simulate such a scenario, we generate two separate configurations: (i) two

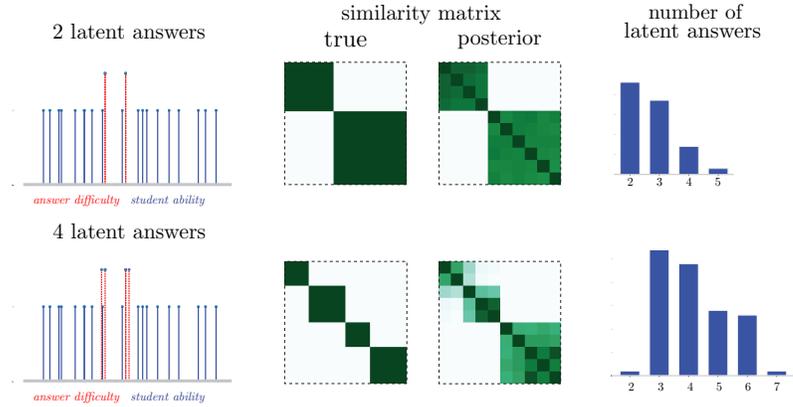


Figure 6.3: Example of inferred posterior distributions for two simulated configurations: 2 latent answers (top panel) and 4 latent answers (bottom panel), where question difficulty parameters  $\{\beta_k\}$  are shown in red, and student ability parameters  $\{s_i\}$  are shown in blue. Two pairs of latent answers in the 4 cluster configuration (bottom panel) have identical parameters (displayed with a small offset for illustration). The true and the posterior distributions over clustering configurations are displayed with a similarity matrix, where the cell  $(i, j)$  in the similarity matrix represents the average fraction of MCMC samples where choice  $i$  and  $j$  belong to the same cluster. Additionally, explicit posteriors over the number of clusters are displayed for each of the two configurations (right-most panel). Our results show that the model is capable of recovering accurate clustering information in a challenging scenario, where two pairs of clusters have an identical set of parameters.

latent answers with distinct parameters ( $\beta = [-1, +1]$  and  $\mathbf{y} = [-1, +1]$ ), and (ii) four latent answers, with two pairs having an identical set of parameters ( $\beta = [-1, -1, +1, +1]$  and  $\mathbf{y} = [-1, -1, +1, +1]$ ). As each configuration only has one question, we drop the  $j$  (question) index in our notation. Additionally, we generate a total of 30 students, with abilities  $s_i$  sampled from a normal distribution  $\text{Normal}(0, 12)$ . The simulated question in our experiment contains 15 choices, and each student observes a random subset of 4 choices on each interaction. A total of 100 interactions are sampled and used for inference.

Figure 6.3 illustrates the inference results. As Gibbs sampling in CRP can result in label-switching (i.e., latent answer assignments may change identity during sampling), aggregating latent answer assignments across samples is impossible without some form of post-processing. To circumvent this problem, we illustrate our sampling results with a similarity (or affinity) matrix instead. A similarity matrix is computed by keeping track of auxiliary variables  $u_{ij} = \delta[z_i = z_j]$  during sampling, where  $u_{ij} = 1$  if the latent answer assignment of choices  $i$  and  $j$  are the same, and  $u_{ij} = 0$  otherwise. The advantage of keeping track of  $u_{ij}$ , as opposed to the explicit latent answer assignments, is because  $u_{ij}$  are invariant to label-switching during sampling. The similarity matrix in Figure 6.3 represents the empirical average of  $u_{ij}$  across all samples. Observe that in both configurations (2 and 4 latent-answers), the model is able to approximately recover the correct number of clusters.

### 6.3.2 Experiment 2: Effect of hyper-parameters

As our model contains an additional hyper-parameter  $\epsilon$ , we perform a series of experiments to study the effect of  $\epsilon$  on the number of inferred clusters as a function of the true number of clusters. The effect of  $\alpha$ , a hyper-parameter of the DP is known and is related to the “propensity” of the model to create new clusters doing inference; larger values of  $\alpha$  will effectively result in smaller but more numerous clusters doing inference. In our model, the hyper-parameter  $\epsilon$  also has a direct impact on the “propensity” of the model to create clusters. Intuitively, the effect of  $\epsilon$  on the number (and the size) of clusters can be understood as follows: the model has a higher posterior probability with configurations that have few clusters (this is clear from (6.4), where the probability of observing a

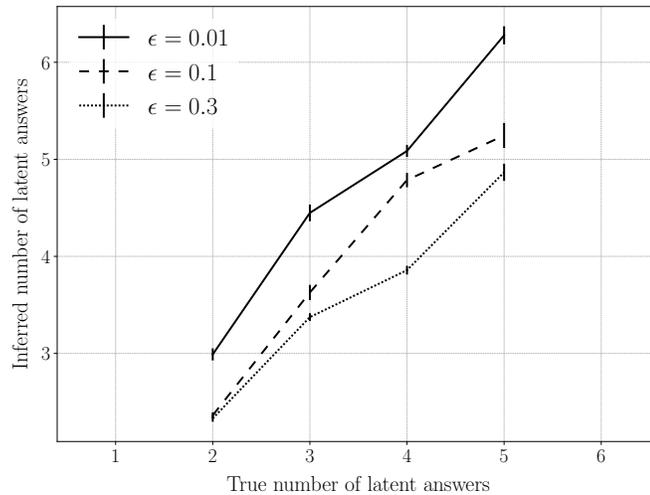


Figure 6.4: Inferred number of clusters (latent answers) vs. true number of clusters as a function of the  $\epsilon$  hyper-parameter (simulation). The proposed model estimates a larger number of clusters for smaller values of  $\epsilon$ .

set of responses is higher when these responses are grouped under the same cluster). The effect of  $\epsilon$  is to effectively allow the model to group choices under the same latent cluster even if some of the students did not select a “consistent” set of choices (i.e., not all of their responses to the choices grouped under a given cluster were identical). Larger values of  $\epsilon$  allow the model to be more tolerant to such “inconsistent” configurations grouped under the same latent answer. As such, we expect that the number of inferred clusters would decrease (and the sizes of the clusters increase) for larger values of  $\epsilon$ . Figure 6.4 plots the average number of inferred clusters as a function of the true number of clusters, averaged across 10 simulations. In each simulation, 10 student ability parameters were sampled from  $\text{Normal}(1, 2)$  and 10 answers (choices) were generated, where each choice was assigned a membership to one of the latent answers (the number of latent answers ranged from 2 to 5 in our experiment and is represented on the the  $x$ -axis in Figure 6.4). Latent answer parameters  $\{\beta_k\}$  were sampled from

Normal(0, 1). Each simulation contained 100 interactions, where an interaction consisted of a student selecting a subset of (what they judged to be) correct answers out of the presented set of 4 random answers (out of 10 total). The value of  $\alpha$  was set to 1 across all runs of the simulation.

A maximum a posterior (MAP) estimate of the number of clusters from each simulation run was computed, averaged across simulation runs for different values of  $\epsilon \in \{0.01, 0.1, 0.3\}$ . Figure 6.4 displays the expected result: (i) the model is able to approximately infer the correct number of clusters for all three values of  $\epsilon$  (up to an error of approximately 1 cluster), and (ii) larger values of  $\epsilon$  result in posteriors with fewer clusters.

## 6.4 Real-World Experiments

We now show two real-world experiments to evaluate the efficacy of our approach.

### 6.4.1 Experiment setup

In order to evaluate the efficacy of the proposed model in a realistic scenario, we conduct two real-world experiments on Amazon Mechanical Turk. Each experiment was carried out in two stages: (i) an *open-response* stage and (ii) a *multiple-choice* stage. In both stages, a group of Mechanical Turk workers (no worker participated in both stages) were recruited to read a chapter excerpt from a textbook. Two units were used: unit 7.2 (Language) from the OpenStax [8] Psychology textbook and unit 15.1 from the OpenStax U.S. History textbook. When a

worker participating in the *open-response* stage indicated that they completed the reading, they were presented with a set of open-response questions, requiring them to type in an answer. When a worker participating in the *multiple-choice* stage indicated that they completed a reading, they were presented with the same set of questions, but in a multiple choice form, with the choices being precisely the open-response submissions of other workers in the *open-response* stage. The workers were instructed to select all correct answers, as well as to ignore minor errors (e.g., spelling or grammar) that do not otherwise affect the answer's correctness.

In total, 15 and 29 workers participated in the *open-response* and *multiple-choice* stages of the Psychology reading respectively, and 24 and 18 workers participated in the *open-response* and *multiple-choice* stages of the U.S. History reading respectively. A total of 15 (Psychology) and 10 (U.S. History) questions were administered in each stage, where most of the questions were extracted from the end-of-the-unit quiz of each textbook, and some were added by the author. An average number of 10.4 (Psychology) and 17.4 (History) open-response submissions per question were shown to each worker in the *multiple-choice* stage, after automatically aggregating options within an edit distance of two (i.e., identical answers and those with small spelling variations).

One of the authors also manually annotated the correctness of each answer in the Psychology reading, which is used as the ground-truth for evaluating the performance of the model in grading the correctness of the open-response submissions. The total number of open-response submissions (across all questions) in the Psychology reading was 151, of which 72 were annotated as correct (52% class balance).

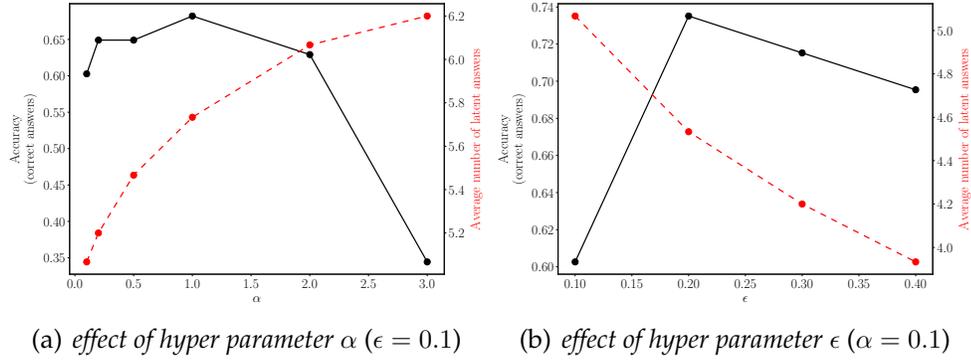


Figure 6.5: Accuracy in predicting correct answers (solid black) and the average number of inferred latent answers per question (dashed red) as a function of hyper parameters  $\alpha$  and  $\epsilon$  (baseline accuracy is 52%). Increasing  $\alpha$  leads to more clusters (latent answers) on average, while increasing  $\epsilon$  leads to fewer clusters on average. Accuracy suffers when the number of inferred clusters is either too few or too many: if (i) the number of clusters is too few, the model potentially groups unrelated answers and (ii) if the number of clusters is too many (in relation to the true number of clusters), the model (incorrectly) treats multiple responses on the answers represented by the same cluster as independent observations, leading to poorer parameter estimates.

In total, the Psychology dataset consists of 435 interactions (student-answer-response tuples). For inference, we run 200 MCMC iterations and discard the first 50 burn-in samples. We compute an estimate of answer correctness based on the kept samples as follows. Let  $\mathbf{y}_{jc} = [y_{jc}^{(1)}, \dots, y_{jc}^{(N)}]$  be a sequence of MCMC samples of the answer's correctness for answer (choice)  $c$  of question  $j$ , where  $y_{jc}^{(n)}$  represents the  $n^{\text{th}}$  sample. Note that while answer correctness  $y_{jk}$  is a cluster-specific (rather than answer-specific) parameter, we can nevertheless choose to maintain an answer-specific correctness  $y_{jc}$  by endowing an answer with the correctness of the latent answer to which it is assigned in the given sampling iteration. We use an empirical average ( $\bar{y}_{jc}$ ) of  $\mathbf{y}_{jc}$  as our final estimate of answer's  $c$  correctness. In computing accuracy, we consider answers with  $\bar{y}_{jc} \geq 0$  as correct, and answers with  $\bar{y}_{jc} < 0$  as incorrect.

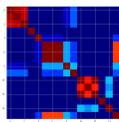
## 6.4.2 Accuracy results

Figures 6.5(a) and 6.5(b) display accuracy (computed as a fraction of the correct predictions, in predicting the correctness of each answer) as a function of the hyper-parameters  $\alpha$  and  $\epsilon$  respectively. From both figures we can conclude that the model is able to correctly predict between 60% and 74% of the answers (with the exception of  $\alpha = 3.0$ ) (recall that baseline performance is 52%).

Recall that both hyper-parameters,  $\alpha$  and  $\epsilon$ , control the model’s “propensity” for creating clusters, which we illustrate by plotting the average number of clusters per question as a function of each hyper-parameter in Figure 6.5 (dashed red). Both plots confirm the expected pattern that we also observed in simulation: increasing  $\epsilon$  creates fewer clusters, while increasing  $\alpha$  creates more clusters. We also observe that accuracy drops in the extreme settings of both hyper-parameters, where the number of cluster is either too few or too many. The drop in accuracy is expected when the number of clusters is too few, as the model may begin to group unrelated answers together. When the number of options is too few, we hypothesize that accuracy suffers as a result of poorer parameter estimates: when the model treats otherwise identical answers as different, a response to each manifestation of an underlying latent answer is treated (incorrectly) as an independent observation, leading to poorer parameter estimates in the model.

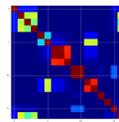
## 6.4.3 Clustering results

We employ the same process described in Section 6.3 for aggregating the MCMC samples of cluster assignments during sampling: namely we keep track of auxiliary variables  $u_{ij} \in \{0, 1\}$  for each pair of answers, and use the empirical



**What was the outcome of the event at Fort Sumter?**

- 0.81 the civil war began
- 0.81 start of the civil war
- 0.81 the kickoff of the civil war.
- 
- 0.45 fort sumter was captured by confederate forces.
- 0.45 confederates fired on it and captured it
- 0.45 the confederacy took over fort sumter.
- 
- 0.36 the attack on fort sumter was seen as the csa declaration of war on the union
- 0.38 this was the initial battle of the civil war
- 
- 0.38 the confederate forces were victorious.
- 
- 1.00 the us won the war
- 0.98 the north won the battle at fort sumter.
- 1.00 the confederates lost the battle to the abolitionist.
- 
- 0.55 it was destroyed



**What is the main difference between a Federation and a Confederation?**

- 0.92 in a confederation states unite for a common purpose but remain autonomous in terms of making their own rules.
- 0.86 in a confederation, individual states join together for a purpose. a federation includes everyone.
- 0.89 the confederation was a more loose grouping - less central government
- 
- 0.79 one is mandatory, other is voluntary
- 0.79 federation membership is not voluntary and confederation membership is
- 
- 0.26 federation believed in freedom while confederation believed in slavery
- 0.40 the beliefs on slavery
- 
- 0.98 i don't recall
- 0.98 i dont know
- 0.95 unsure.

Figure 6.6: Answers (open-response submissions) in response to two questions from an OpenStax U.S. History textbook. Dashed lines group latent answers according to the results of the inference. Numbers next to each answer represent inferred answer correctness (answers with values  $\geq 0$  can be interpreted as correct answers, and as incorrect answers otherwise, highlighted in red). A similarity (affinity) matrix (described in Section 6.3) is shown for each question.

average of  $u_{ij}$  across samples to estimate the “affinity” of the two answers (i.e., a fraction of samples that choices  $i$  and  $j$  belong to the same cluster). Although the similarity (or affinity) matrix is useful for aggregating the samples and visualizing the posterior, it is also valuable to obtain a single, representative clustering configuration from the posterior. We use spectral clustering (with the number of clusters set to the MAP estimate from the posterior on the number of clusters) as a simple post-processing step on the similarity matrix to obtain such a clustering configuration.

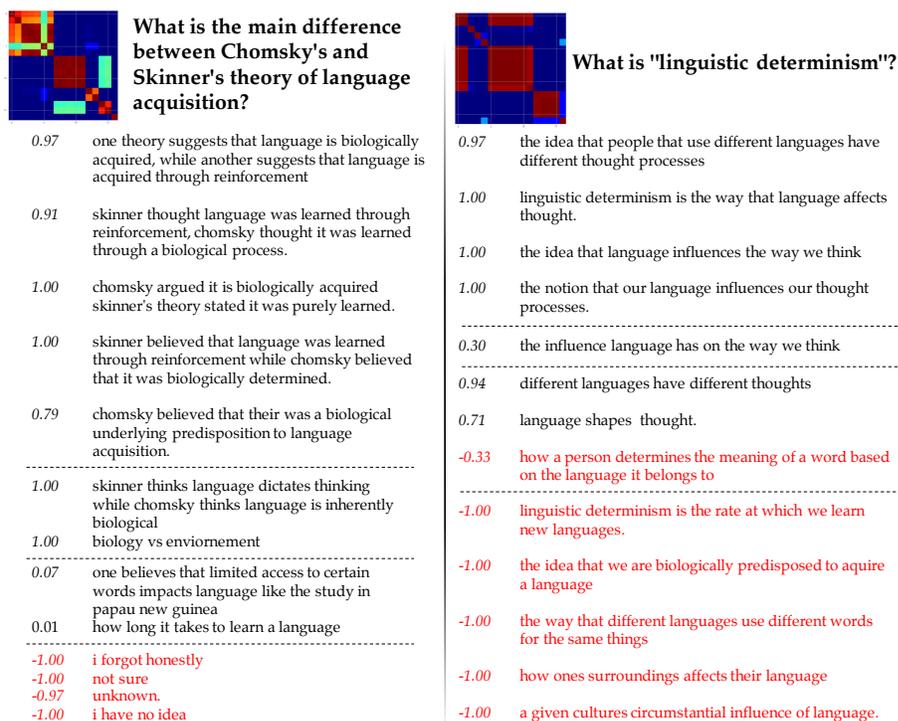


Figure 6.7: Answers (open-response submissions) in response to two questions from an OpenStax Psychology textbook. See caption of Figure 6.6 (above) for details.

We illustrate the clustering performance via examples in Figures 6.6, 6.7 and 6.8. Figures 6.6 and 6.7 each show 2 questions from the U.S. History and Psychology textbooks respectively. Each question lists most of its answers (several were removed due to space constraints, but ensuring that the remaining answers are representative of the entire set). Dashed horizontal lines separating groups of answers are obtained from the output of the spectral clustering step described above. Numbers next to each answer represent  $\bar{y}_{jc}$  the empirical average of  $y_{jc}$  (choice correctness), averaged across MCMC samples, which is in the range  $[-1, 1]$ . As done in Section 6.4, we consider answers with  $\bar{y}_{jc} \geq 0$  as correct, and answers with  $\bar{y}_{jc} < 0$  as incorrect. Answers predicted to be incorrect are highlighted in red.

By what age does a child typically start to converse in complex sentences (more than a few words)?

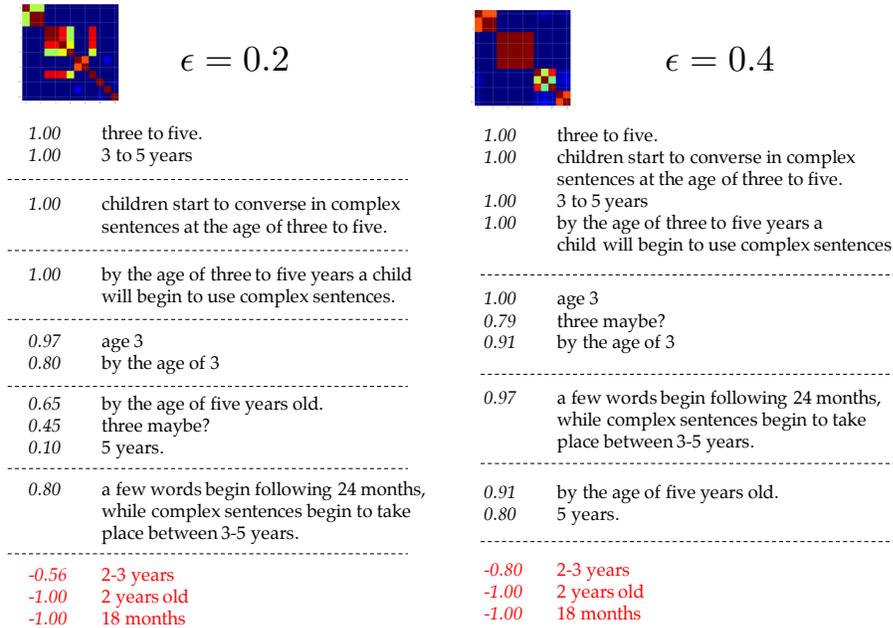


Figure 6.8: Effect of the hyper-parameter  $\epsilon$  on the inferred cluster for a question from an OpenStax Psychology textbook. As expected, we observe that a larger value of  $\epsilon$  results in fewer (and larger) clusters. See caption of Figure 6.6 for additional details on interpreting the figure.

Both Figure 6.6 and Figure 6.7 demonstrate that the model is able to group together semantically identical (or similar) items, e.g., “start of the civil war,” “the kickoff of the civil war,” and “the civil war began” (top left of Figure 6.6). While natural language processing (NLP) techniques are likely capable of identifying such answers as paraphrases (by relying on external resources), our model is able to identify close semantic relatedness among answers without inspecting the content.

Some answers are also misgrouped by the model, either (i) resulting in unrelated answers grouped together or (ii) the model failing to recognize a group

of related answers and attributing them to distinct clusters. For example, in Figure 6.7 (bottom right), while some of the incorrect answers that are grouped together are semantically similar, many are not (e.g., “rate at which we learn language” vs. “culture’s influence on language”). The cause of this error stems from the fact that these incorrectly-grouped answers are “too incorrect”, i.e., almost no student had chosen them as correct in our dataset because they were too easy to rule-out. As a result, from the perspective of the model, there is no signal to indicate that the answers are in fact distinct. The key feature that makes our model domain-agnostic, i.e., relying only on the statistics of co-selection among answers, also makes it highly dependent on the sufficient amount of observations (of answer selection) that also contain enough variability to reveal the underlying latent structure. At the same time, it is important to note that in practice, a purely domain-agnostic solution such as the one we propose in this work, will likely benefit from the addition of a domain-specific signal as well (e.g., text of the answers), which would be complementary in the regime of insufficient observations (such as our example in Figure 6.7).

Figure 6.8 displays the inferred clustering configurations for the same question, but with the inference performed for two different settings of the hyperparameter  $\epsilon$ . The results support the expected effect of  $\epsilon$  on the number of clusters: larger  $\epsilon$  (right panel in Figure 6.8) result in fewer (and bigger) clusters. We can see that a larger  $\epsilon$  ( $\epsilon = 0.4$ ) identifies all clusters correctly, while a smaller  $\epsilon$  ( $\epsilon = 0.2$ ) breaks up some of the otherwise semantically-related groups of answers (e.g., “three maybe?”). An intuitive explanation for many of such “splits” (for a lower  $\epsilon$ ) is the following. Although the answers might be semantically-similar, they are very rarely identical. Recall from Section 6.4.1 that the students (workers) are mainly instructed to select all *correct* answers, and on the basis of these selections

the model attempts to discover those answers that are similar. If the answers are sufficiently different in the student’s mind to an extent that it affects their judgement of the answers’ correctness, to the model, this is a signal that the answers do not belong in the same cluster. Although only a minority of the students may judge the answers as distinct, the extent of the impact of that minority’s judgement on the inference is directly controlled by the hyper-parameter  $\epsilon$ .

## 6.5 Related Work

Our work resides at the intersection of three research directions: *student submission clustering*, *peer-grading*, *crowdsourcing*. We briefly summarize relevant work in each research area and highlight its relevance to our contribution.

### 6.5.1 Student submission clustering

There has been a significant amount of progress in developing methods and interfaces for effectively grouping and grading similar student submissions (assignments) in multiple domains, including short-answers, mathematical expressions and computer code. Most of the methods for clustering text-based submissions (e.g., short answers) employ various similarity functions to measure the similarity between different submissions, and use machine learning methods to learn the importance of various features in computing the similarity. The goal is to learn similarity functions that detect semantic similarity between submissions despite lexical and syntactic variations. Representative work in this area includes [9, 64, 130, 24]. In contrast to methods that learn a similarity function,

other approaches such as [106] and [86] explicitly encode rules, transformations (e.g., spelling correction) and paraphrases to group related answers. A general consensus in literature is that clustering submissions before grading reduces grading time.

There has also been significant effort in devising methods for clustering other types of submissions, such as computer code [207, 60, 185] and mathematical expressions [103]. Generally, these techniques also rely on “featurizing” the content (e.g., syntax tree of code) to aggregate similar submissions.

In contrast to these methods, our approach is capable of clustering semantically related submissions without analyzing the content of the submissions, thus being domain-agnostic. In practice, however, we expect that a combination of our approach and domain-specific methods will result in superior performance to either method in isolation.

## 6.5.2 Peer-grading and crowdsourcing

Our model shares many of the aspects of related models deployed for the tasks of *peer-grading* [95] and *crowdsourcing* [82] in general. In peer-grading, a fundamental challenge that has been addressed by recent work is that of *grade aggregation*, i.e., coming up with an optimal final score or ranking based on the input from multiple peers. Probabilistic models for grade-[143] and rank-aggregation [148] explicitly model grader bias and reliability, which are estimated from observation, and used in obtaining the final score (or rank).

Peer-grading can be viewed as a variant of a more general task, known as *label*

*aggregation*, which is concerned with a problem of estimating some ground-truth (e.g., grade in peer-grading) from multiple noisy “labelers” (e.g., graders in peer-grading). Refer to [82] for an overview of this field. Probabilistic models such as [199, 7] explicitly model the ability of the “labelers” and the difficulty of the items being labeled to estimate some underlying ground truth (label). Our work can be viewed as a generalization of these methods, that in addition to learning the ability and difficulty parameters, also discovers clusters among similar items.

### **6.5.3 Independence of irrelevant alternatives**

Fundamentally, the model proposed in this chapter addresses the violation of the *independence of irrelevant alternatives* assumption (IIA) [153, 114] in the setting of multiple choice testing. The IIA assumption, implicitly encoded in discrete choice models such as the multinomial logit, postulates that the ratio of probabilities (probability of choosing) of any pair choices is unaffected by the introduction of additional choices. This assumption is violated whenever a subset of the presented choices are identical or nearly identical to the user (e.g., two options in the multiple choice question are semantically identical). A large body of work exists that addresses the IIA limitation of the traditional multinomial logit model that explicitly accounts for a grouping of identical alternatives in discrete choice models. One of the most well-studied extensions is the nested multinomial logit model [14], which posits that choices (alternatives) are grouped into nests (that may be nested in an arbitrarily deep hierarchy), and that the user can be thought of as first deciding among which nest to choose (regardless of which items are in the nests), and then choosing an item out of that nest (introducing additional items into the same nest does not affect the

probability of the user selecting an item out of a different nest). There have been a number of extensions [197, 35, 93, 193, 22] of the original nested logit model proposed since its original introduction by McFadden [14], however, until very recently there has been no work in attempting to learn the nesting structure (all nested logit models assumed that the expert either provided the structure or performed model selection to identify it). The only work that we are aware of that learns the nested structure was developed concurrently and published in the April of this year (2016) [120]. The contribution of their work is a quadratic-runtime algorithm for inducing a nesting structure based on a series of hypothesis tests to identify related items. Our work also learns a structure that groups items (a clustering configuration of one level) based on the observed choices, but is richer in that it also integrates user-specific parameters (ability). Another model that captures non-independence of alternatives is a conditional probit model that uses a joint distribution with a full covariance matrix to model correlations among choices [189]. This model, however, requires learning  $O(n^2)$  parameters (where  $n$  is the number of choices or items), i.e., does not assume an underlying low-dimensional structure among groupings. This assumption (our model) facilitates parameter estimation, interpretability (relevant in clustering students' submissions), and optimal choice set design.

## 6.6 Conclusion and Future Work

In this chapter, we have developed a novel probabilistic framework for simultaneously *clustering* and *grading* open-response submissions to questions on an aptitude test—two key challenges that massive open online courses (MOOCs) face today. In contrast to existing approaches for clustering student submissions,

our framework is domain agnostic, which we achieve by leveraging other students to *implicitly* grade and cluster open-response submissions through a task that is disguised in the form of a multiple-choice test. As a result, our framework has two key advantages over existing methods: (i) it does not require domain-specific solutions for clustering open-response submissions and (ii) by disguising the task of grading and clustering as additional testing, our framework offers a natural way for scaling open-response assessment to massive classrooms—one of the long-standing goals of MOOCs.

In a practical deployment, however, we believe that the best-performing approach will be one that leverages both, the structure of the domain and the intelligence of the crowd, in simultaneously discovering groups of related submissions and grading them. The non-parametric Bayesian approach proposed in this chapter is sufficiently flexible to accommodate a natural integration with methods that exploit domain knowledge for discovering related submissions. Recently proposed distance dependent Chinese restaurant process (DDCRP) [19], for example, offers a natural way for integrating a distance function between pairs of items (submissions) into a conventional CRP-based mixture. We believe that developing models that integrate both sources of information in a principled manner opens a fruitful direction for future research.

## CHAPTER 7

### CALIBRATED SELF-GRADING

#### 7.1 Introduction

Peer-grading is widely believed to be an inexpensive and scalable way to assess students in large classroom settings. In this chapter, we propose *calibrated self-grading* as a more efficient alternative to peer grading. For self-grading, students assign themselves a grade that they think they deserve via an incentive-compatible mechanism that elicits maximally truthful judgements of performance. We show that the students' self-evaluation scores obtained via this mechanism can be used to perform classic item response theory (IRT) analysis. In order to obtain unbiased estimates of the IRT parameters, we show that the self-assigned grades can be calibrated with a minimum amount of input from instructors or domain experts. We demonstrate the effectiveness of the proposed calibrated self-grading approach via simulations and experiments on Amazon's Mechanical Turk.

A significant bottleneck in scaling traditional classrooms to hundreds or thousands of students is the challenge of enabling efficient mechanisms of assessment. Peer-grading, hailed as a solution to this "scaling problem," has received significant attention, both from the education [162, 95] and machine learning [143, 148] communities. Broadly speaking, peer-grading can be thought of as a relaxation of the traditional teacher/student roles in the classroom: An expert instructor is replaced by several "noisy" students having the task of estimating performance of other students. Virtually all of the existing statistical models for peer-grading

---

<sup>0</sup>This chapter has been adapted from the paper [100]

aim to estimate the student's true performance from such noisy measurements, under some metric of optimality.

*Self-grading* constitutes a special case of peer-grading: The student is their own only "peer" and is solely responsible for assigning a score based on the judgement of their own work. Depending on the student's honesty in self-evaluation, self-grading is appealing for at least two reasons: (i) Students can provide a richer signal towards their internal state of knowledge by explicitly revealing confidence in their answers—a signal that can be exploited during assessment; (ii) because every student is their own grader, potentially no additional peer-grading efforts are required to perform assessment. Self-grading, however, introduces two unique challenges not faced in traditional peer-grading: (i) Designing mechanisms for eliciting honest judgement of performance and (ii) accounting for individual biases in self-evaluation. The first challenge in self-grading fundamentally requires an explicit mechanisms for eliciting truthful judgements.<sup>1</sup> The second challenge is addressed in peer-grading by appealing to statistics and assuming that the population of graders is—at least on average—unbiased.

In this chapter, we propose *calibrated self-assessment* to address both of the above challenges. Our approach combines self-assessment with a small number of instructor-graded items, which provides a simple, incentive-compatible mechanism of eliciting self-assigned scores, and yields assessments of comparable or superior quality to a setting with significantly more instructor-graded items and no self-scoring. As a consequence, calibrated self-assessment enables a significant reduction in effort of instructors, domain experts, or peers.

---

<sup>1</sup>This is also a potential problem in peer-grading when conflicts of interest are present.

## 7.2 Related Work

We focus our review on two research directions that the work in this chapter aims to bring together: (i) self-assessment as a method for summative assessment and (ii) decision-theoretic mechanism design for judgement elicitation.

**Self-grading and Peer-grading in education:** Self-assessment is often seen by teachers as a valuable tool in classrooms [188], who cite self-assessment as a viable way to reduce the instructor's effort, elicit additional information from students (e.g., their effort and confidence), and provide an additional learning opportunity in the process. More recently, in addition to peer-grading, self-grading was deployed in massive open online courses (MOOCs) [95]. Self-grading as a tool for summative assessment, however, is controversial, with its validity questioned on the basis of students' internal biases. In fact, studies indicate that bias is often a function of one's ability [188, 182]. Studies that compare peer-grading and self-grading differ in their findings, with self-grading and peer-grading performance excelling in different conditions (classrooms, age-groups, etc.), but both are heavily influenced by the underlying assessor biases (see [182] for a survey of the studies). A study carried out in four middle-school science classrooms found that peer-grading and self-grading have a high correlation with instructor grades, with grading bias patterns that are consistent with other studies [162]. In addition, they found that the process of self-grading resulted in learning gains, whereas peer-grading did not. A recent study carried out at the university level, however, found that both peer-grading and self-grading results in learning gains as a side-effect of grading [138].

The existing literature on self-grading points to the significant effect of bias

in self-scoring, with most studies concluding that students of lower ability tend to inflate their grades more. As a consequence, we argue for the importance of an incentive-compatible mechanism that is designed to elicit maximally truthful judgements, and a *calibrated* model that is able to explicitly de-bias the individuals by incorporating a subset of instructor-graded items.

**Judgement elicitation:** The literature on truthful judgement elicitation through scoring functions dates back to the fifties, when the so-called “quadratic scoring rule” was proposed for the task of weather forecasting [23]. Since then, a number of generalizations of the quadratic scoring rule and other incentive-compatible scoring rules have been proposed and analyzed [62, 172, 133, 165] and found application in forecasting weather, sports, and finance. Analysis of the behavior of non-risk neutral agents in scoring-rule-based mechanisms has received only limited attention [142], with lottery-based payoffs being the most well-known solution for encouraging risk-neutral behavior. Lottery-based payoffs had received mixed results in experimental evaluations [72, 173], and in the context of education a reward system based on a lottery is not a reasonable solution. In this chapter, we rely on heavily limited instructor input in order to correct for individual biases, which includes under- and over-confidence, as well as non-risk-neutral behavior.

To the best of our knowledge, the only work that applies a scoring rule mechanism in the context of education that we are aware of is [16]. The focus of their work is in analyzing the effect of the different scoring functions on the self-assessment behavior of students. Our primary contribution in this chapter is in developing a principled statistical model for calibrated assessment that integrates self-scoring and instructor-scoring within the classic IRT framework.

### 7.3 Model

Self-grading without a proper incentive mechanism may lead to dishonest behavior. In the setting of self-grading, a “mechanism” is a *scoring rule* that specifies the rules by which the points are assigned to the student as a function of their own judgement and the outcome (i.e., whether their answer was correct). A mechanism is called *incentive compatible* when the student’s optimal strategy with respect to his or her own utility function results in a truthful elicitation of information, e.g., truthful judgement of their own work.

We consider the following scoring function:

$$p_{ij} = \begin{cases} \theta_{ij} & \text{if correct} \\ -\frac{1}{2}\theta_{ij}^2 & \text{if wrong,} \end{cases}$$

where  $\theta_{ij} \in [0, A]$  is a score provided by student  $i$  in answering question  $j$ , where  $A$  is some fixed upper bound. If the student provides a correct answer, they get the  $\theta_{ij}$  points that they proposed; if they provide an incorrect answer, they lose exactly half of that value squared. This scoring function is known as a *quadratic scoring rule* and was first proposed in [23].

For this scoring function, the expected payoff is

$$\mathbb{E}[p_{ij}] = \theta_{ij}\hat{\pi}_{ij} - \frac{1}{2}\theta_{ij}^2(1 - \hat{\pi}_{ij}), \quad (7.1)$$

where  $\hat{\pi}_{ij}$  is the  $i^{\text{th}}$  student’s estimate of the probability that they will get question  $j$  correct. This expression is maximized when

$$\theta_{ij} = \frac{\hat{\pi}_{ij}}{1 - \hat{\pi}_{ij}}. \quad (7.2)$$

Equation 7.2 is exactly the student’s own belief about the odds of them answering the question correctly. Consider that the student estimates their chances of

answering any question correctly, by simultaneously estimating their own ability and the difficulty of the question. Let us now define that probability to be the standard IRT Rasch likelihood, but defined with respect to the student's own estimate of their ability,  $\hat{s}_i$  and their estimate of the question's difficulty  $\hat{q}_j$ :

$$\hat{\pi}_{ij} = \frac{1}{1 + \exp(-(\hat{s}_i - \hat{q}_j))}.$$

Given the student's estimate of their own ability  $\hat{s}_i$  and of the difficulty of the question  $\hat{q}_j$ , we can now derive their optimal proposed score (assuming they act rationally and are risk-neutral) for that problem  $\theta_{ij}$  (or rather its logarithm):

$$\log(\theta_{ij}) = \hat{s}_i - \hat{q}_j,$$

which follows from the fact that log-odds of a logistic model is a linear function of its parameters. We will assume that the student is risk-neutral and is unbiased in his or her estimates of own ability and question difficulty, but we will relax both assumptions later. On any given question, however, the student's estimate of their ability to answer that particular question may deviate from their true ability. Assuming that the student's own estimates are normally distributed around their true values, we get:

$$\hat{s}_i - \hat{q}_j \sim \mathcal{N}(s_i - q_j, \sigma^2),$$

where  $s_i$  and  $q_j$  are the true student ability and question difficulty parameters respectively. As a consequence, it follows that  $\log(\theta_{ij})$  is normal distributed and  $\theta_{ij}$  is log-normal distributed. Consider a dataset  $D$  consisting of the self-assigned scores  $\log(\theta_{ij})$  submitted by each student for each question that the student answered. We can write the conditional likelihood of the entire dataset as follows:

$$P(\boldsymbol{\theta} \mid \mathbf{s}, \mathbf{q}) = \prod_{(i,j) \in D} \mathcal{N}(\log(\theta_{ij}) \mid \mu = s_i - q_j, \sigma^2).$$

Here,  $\mathbf{s}$  and  $\mathbf{q}$  are the vectors comprising the student ability and question difficulty parameters, respectively, and  $\boldsymbol{\theta}$  is the vector of student-submitted scores. Maximizing the likelihood of all observations gives a straightforward least-squares solution for the parameters  $s_i$  and  $q_j$ , given all the user-provided scores  $\theta_{ij}$ . Note that  $\sigma^2$  is assumed to be a constant variance in students' estimates of their own ability. In practice this variance is likely user-specific and corresponds to the students' ability in self-assessment. We will address the issues of bias and variance in self-assessment in Section 7.3.2.

### 7.3.1 Parameter estimation

It is interesting to note that we can solve for the IRT parameters (student abilities and question difficulties) using the above formulation with *no* outcome information, i.e., without knowing which students answered which questions correctly. In fact, the above approach does not even require that the students who are self-grading know what the correct answer is; students' confidence in their answers elicited through the quadratic scoring rule is all that is needed to learn the parameters of the model. Of course, this observations relies on two fundamental assumptions: (i) students are risk-neutral and (i) students are unbiased in estimating their chance of answering a question correctly. In Section 7.3.2, we will account for the individual biases and non-risk-neutral behavior by explicitly introducing a bias parameter into the model and estimating it from an additional set of instructor-graded responses. However, in order to gain a better understanding of the model, it is insightful to first analyze the solution to the problem where both of these assumptions hold.

The solution for the model parameters can be obtained in closed-form using a standard pseudo-inverse solution to a least-squares problem. Alternatively, the solution can be obtained iteratively, without requiring to explicitly invert any (potentially large) matrices. In particular, one can repeatedly evaluate the following two steps:

$$s_i = \sum_{j \in Q_i} \frac{q_j}{\lambda + n_q^i} + \sum_{j \in Q_i} \frac{\log(\theta_{ij})}{\lambda + n_q^i}$$

$$q_j = \sum_{s \in S_j} \frac{s_i}{\lambda + n_s^j} - \sum_{i \in S_j} \frac{\log(\theta_{ij})}{\lambda + n_s^j}.$$

Here,  $s_i$  is the ability of student  $i$  and  $q_j$  is the difficulty of question  $j$ . To guarantee a unique solution, we introduce a non-negative regularization parameter  $\lambda$ , which we will discuss in more detail in the next paragraph. The constants  $n_q^i$  and  $n_s^j$  are the number of questions that student  $i$  answered and the number of students that answered question  $j$  respectively. Note that the above iterative solution has an intuitive interpretation: The ability of the student is the sum of the average of the (log-transformed) self-assigned scores to a set of questions that the student answered and the average difficulty of those questions. In turn, the difficulty of a question is the negative of the average (log-transformed) score that students assigned to themselves for that question plus the average ability of the students who answered that question. Intuitively, if students with high ability self-assess themselves to have done poorly on a specific question, that question will have a large difficulty parameter.

In the case where there is no missing data, i.e., each student answers each

question, the solution for student ability parameters simplifies to:

$$\mathbf{s} = \begin{bmatrix} \frac{\sum_{i \in S} \log \theta_{i1}}{\lambda + N_s} \\ \vdots \\ \frac{\sum_{i \in S} \log \theta_{iN_q}}{\lambda + N_s} \end{bmatrix} + \mathcal{O}(1/\lambda)\mathbf{1},$$

where  $\mathcal{O}(1/\lambda)$  is a function that grows proportional to  $1/\lambda$ . In other words, the student's ability is simply the average of the (log-transformed) scores that the student assigned to themselves plus a constant that is identical for each student. This solution also illustrates the role of the regularization parameter  $\lambda$ . Because the solution for  $\mathbf{s}$  and  $\mathbf{q}$  is location-invariant, without an explicit prior, the likelihood is maximized by scaling all parameters to infinity. This is equivalent to setting  $\lambda$  to 0, in which case the above solution will tend to infinity, as expected. Note, however, that the relative ranking of the student abilities in this solution will be consistent, regardless of  $\lambda$ . As obtaining the ranking of the students is our primary focus, we can thus set  $\lambda$  to zero in the above solution, and simply consider the average self-assigned (log-transformed) score as the the ability parameter of the student. The same argument applies to question difficulty parameters.

### 7.3.2 Calibrating the model

There are two issues in relying on students' self-given score for ranking students via the IRT model: (i) Students may be prone to over- or under-estimating their ability and (ii) because there is uncertainty involved in both answering and grading, some students may be more or less inclined to "gamble" with their self-assigned score (i.e., some students are more or less risk-averse/risk-loving). We subsume both effects (as it is impossible to tell them apart) into a general

student “bias” in self-grading, and model it explicitly as

$$\log(\theta_{ij}) = \hat{s}_i - \hat{q}_j + b_i,$$

where  $b_i \in (-\infty, \infty)$  is a student-specific bias. We assume that this student bias is drawn from a normal distribution  $b_i \sim \mathcal{N}(0, \sigma_b^2)$ , where the above distribution stipulates that the average of the student population is unbiased. It is impossible to estimate  $b_i$  using self-grading alone, as without actual observations of correctness of students’ responses, the model will conflate  $s_i$  and  $b_i$  into a single parameter. Imagine that we do grade a student’s responses on a small subset of the answered questions (which they also self-grade). Let the set of instructor-graded questions be  $Q_g \subseteq Q$ , where  $Q$  is the set of all questions. As the observations of instructor- and self-assigned grades are all conditionally independent given the student and question parameters, the overall likelihood of both self- and instructor-given scores is a product of these likelihoods. We can then express the log-likelihood of the entire dataset as a sum of the self-graded response log-likelihoods and instructor-graded response log-likelihoods:

$$\begin{aligned} \log P(\boldsymbol{\theta}, \mathbf{y} \mid \mathbf{s}, \mathbf{q}, \mathbf{b}) = & \sum_{s_i \in S} \left( \underbrace{\sum_{q_j \in Q} (\log \theta_{ij} - (s_i + b_i - q_j))^2}_{\text{self-graded responses}} \right. \\ & \left. + \underbrace{\sum_{q'_j \in Q_g} \log(1 + \exp(-y_{ij}(s_i - q'_j)))}_{\text{instructor-graded responses}} \right). \end{aligned}$$

Here,  $y_{ij} \in \{-1, 1\}$  is the instructor-grade for question  $j$  answered by student  $i$  and  $\mathbf{y}$  is the response vector for all students ( $y_{ij} = +1$  corresponds to a correct response and  $y_{ij} = -1$  otherwise). Observe that the “bias” parameter only appears in the self-graded part of the likelihood. This allows us to calibrate the model via instructor-graded questions as a “training set” to separate the effects of the bias and true ability. Note that, unlike in the previous case that relied

entirely on students' self-scores, like with the traditional Rasch IRT model, we are unaware of a closed form solution for this formulation. In all of our experiments, we use the L-BFGS algorithm [212] for learning model parameters.

### 7.3.3 Consequences of students' awareness of the mechanism

The assumption that the learner is optimizing a utility function based on the expected test score:

$$\mathbb{E}[p_{ij}] = \theta_{ij}\hat{\pi}_{ij} - \frac{1}{2}\theta_{ij}^2(1 - \hat{\pi}_{ij}) \quad (7.3)$$

fundamentally assumes that the student believes that each question will be graded, as otherwise there would be no possibility of getting a question wrong and losing points. In practice, our goal for self-grading may be motivated by the effort to reduce the instructor's involvement in grading, and, in general, as a way to scale assessment to potentially very large classrooms, such as massive open online courses (MOOCs). Having each submission be graded by an instructor (or your peers) defeats the purpose of self-grading. If, however, the student is aware of the fact that not every question is graded, we can expect that their utility function, and thus their optimal strategy, will be affected by this knowledge. If the test is administered once, of course, the students could be deceived into believing that every question is graded. In a real course, however, a more realistic assumption is that the students possess the knowledge that not all of the questions are graded and if the assignments are returned, we can expect that the students' estimates of the fraction of graded questions will improve over time. If, however, the student believes that a random subset of their submissions is graded by someone else, but if the student does not know which subset is graded, then we should still expect the student's optimal behavior to be maximizing a

utility function similar to the one above. The utility function will not be the same, as we now have to account for the student's belief about how many problems are graded by someone else. Let us assume that the student has a prior belief that each problem has a probability  $\rho$  of being graded. Then, the expected score the student  $i$  receives on question  $j$  is given by

$$\mathbb{E}_{gr} [\mathbb{E}[p_{ij} \mid \text{graded}]] = \rho(\theta_{ij}\hat{\pi}_{ij} - \frac{1}{2}\theta_{ij}^2(1 - \hat{\pi}_{ij})) + (1 - \rho)\theta_{ij},$$

where we take an additional expectation with respect to the student's belief that the problem is graded. Note that when a problem is *not* graded, the expected score that the student receives is just  $\theta_{ij}$ , i.e., their self-assigned score, regardless of whether the student answers correctly. This is because when a problem is not graded, there is no possibility of losing points. We can show that the student's optimal self-assigned score  $\log(\theta_{ij})$  has the following approximate relationship to their ability and question difficulty (the approximation is a piece-wise linear approximation to the true strategy that is asymptotically accurate):

$$\log(\theta_{ij}) = \max \left\{ \log\left(\frac{1}{\rho} - 1\right), (s_i - q_j) - \log \rho \right\}.$$

The optimal strategies for different values of  $\rho$  are illustrated in Figure 7.1. The student's knowledge of the mechanism is reflected by the appearance of a lower-bound on the self-assigned score in a region where the student is likely to do poorly (low values of  $s_i - q_j$ ). This is expected: If the student is aware that the chance of a particular question to be graded is low enough, it would make sense to take advantage of those odds and "bet" a small, but a non-zero amount, even if the student does not know the correct answer. From a practical perspective of implementing a system that solicits self-assessment scores, it would not make sense to provide the user with the ability to provide a self-assessment score lower than their optimum. From the model inference perspective, this introduces a

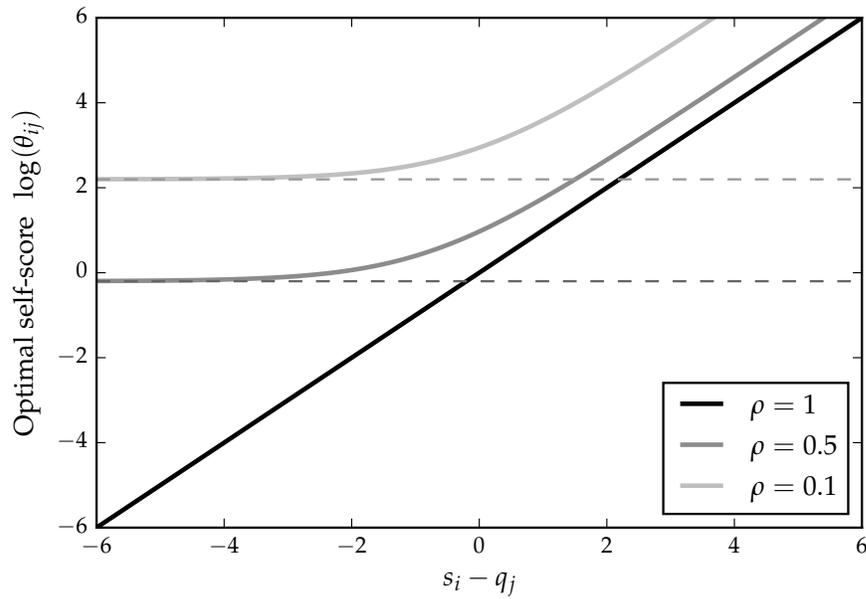


Figure 7.1: The optimal strategy for providing a self-assessment score  $\log \theta_{ij}$  for a student with ability  $s_i$  on a question of difficulty  $q_j$ , assuming the student's knowledge that a random fraction  $\rho$  of the questions will be graded. The optimal strategy is approximately piece-wise linear as a function of the student's relative ability  $s_i - q_j$ . In the regime of low relative ability, the student's optimal strategy is to report a fixed score that is a function of  $\rho$ , regardless of his or her relative ability.

complication: Observations that correspond to the lowest possible self-score do not correspond to any specific  $s_i - q_j$ , but rather an entire range. This problem is known generally as *censored regression*. and can be solved using the same approach as for the original problem, but with the modified likelihood function that accounts for this "kink." Note that a similar restriction on the likelihood (but as an upper-bound) is introduced when the maximum attainable score for a problem is incorporated into the scoring function.

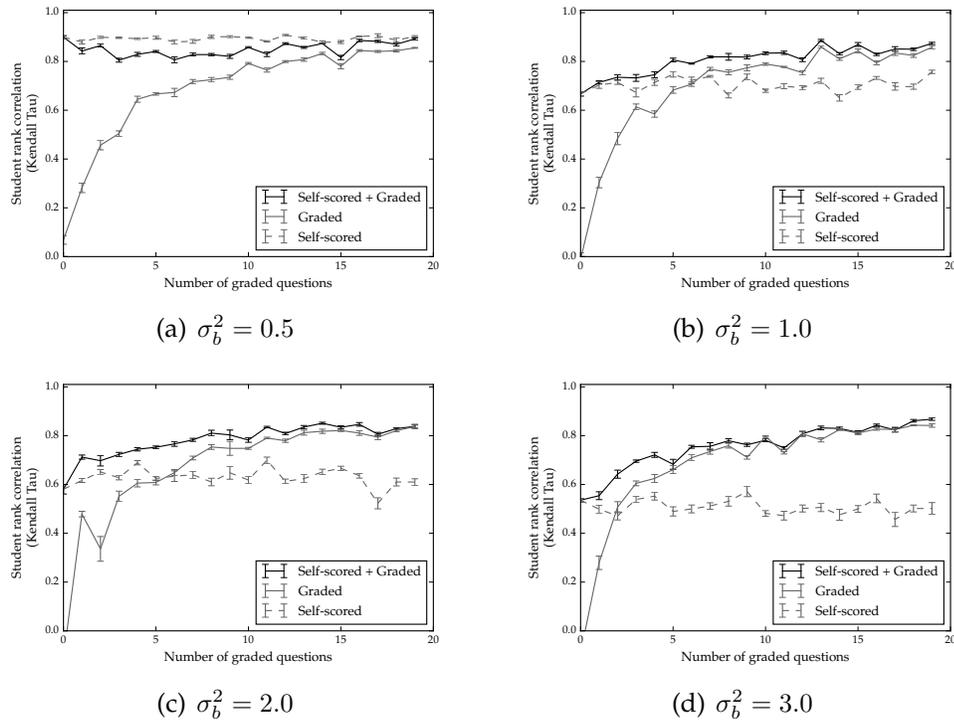


Figure 7.2: Simulation results. Rank correlation across students obtained using three models for different variance of self-grading bias ( $\sigma_2$ ): (i) *black*: a model that uses student self-scores and the correctness of their response to a subset of graded questions (number of graded questions on  $x$ -axis), (ii) *solid gray*: a model that uses correctness of their response to a subset of graded questions only (number of graded questions on  $x$ -axis) and (iii) *dashed gray*: a model that uses only the students' self-score.

## 7.4 Experiments

### 7.4.1 Simulations

It is insightful to study the effect of bias in the population of students on the quality of the learned parameters in the IRT model: student ability parameters and question difficulty parameters. We perform a simple simulation of a classroom with 50 questions and 30 students (question difficulties and student abilities are

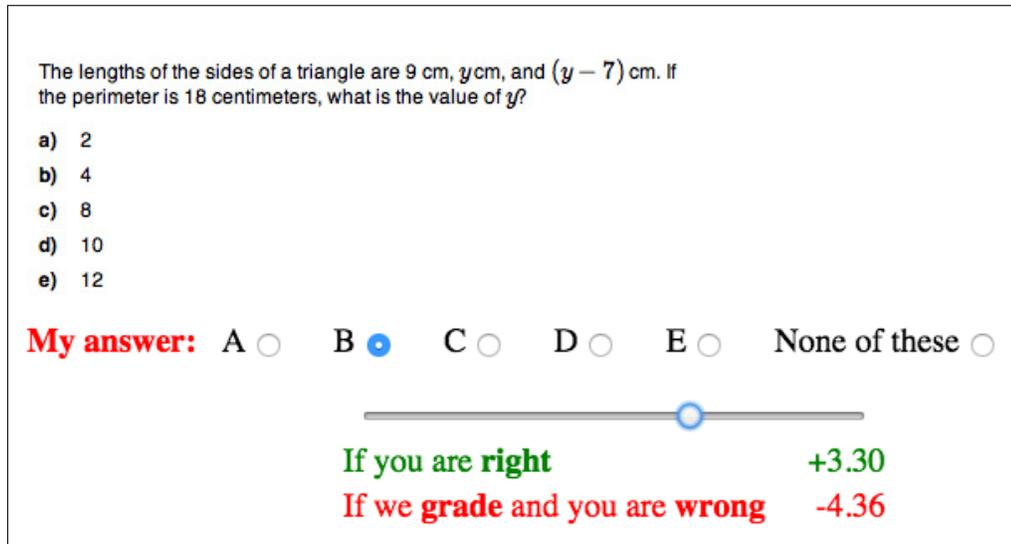


Figure 7.3: Screenshot of one question from the Mechanical Turk task. A subject answers a math question and provides a self-assessment score by adjusting a slider. The student sees the number of points that they will gain if they answer the question correctly (green) and the number of points they will lose if they answer the question incorrectly (red).

sampled from a zero-mean normal distribution with a standard deviation of 3), where each student answers each question (a total of 1,500 responses). In this simulation, each student submits their self-grade  $\log(\theta_{ij})$  for each question by optimizing their utility according to the utility function in 7.3. We repeat the simulation for four different populations of students, each with a different variance  $\sigma_b^2$  of the bias parameter. To evaluate the quality of the inferred student parameters, we compute the rank correlation (Kendall Tau) between the true ordering of the students (by their true parameters) and the ordering obtained by sorting the students based on the inferred parameters. The Kendall Tau metric is defined as follows:

$$KendallTau(\mathbf{s}, \hat{\mathbf{s}}) = \frac{N_{\text{pairs}}^{\text{correct}} - N_{\text{pairs}}^{\text{wrong}}}{N_{\text{pairs}}}$$

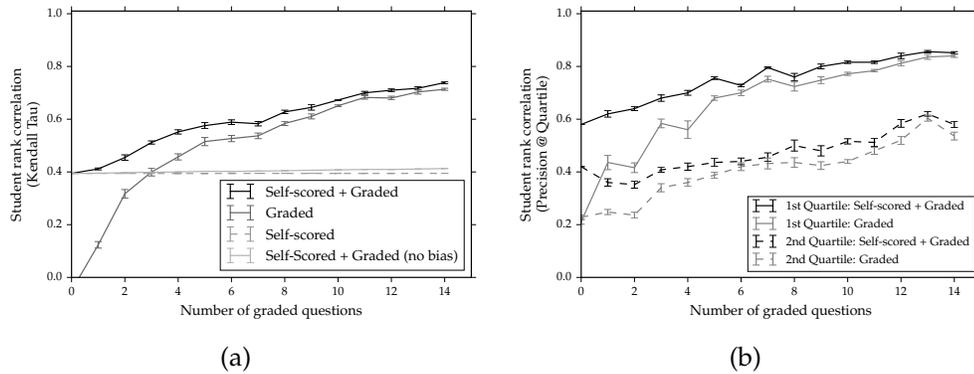


Figure 7.4: User study results. Rank correlation across students obtained using three different models (i) **Self-scored**: a model that relies entirely on student-submitted self-assessments, (ii) **Graded**: a model that relies entirely on instructor-provided grades, as a function of the number of graded questions ( $x$ -axis), and (iii) **Self-scored + Graded**: a model that aggregates students' self-assessment scores on all questions and a variable number of instructor-graded questions ( $x$ -axis). (a) Computes rank correlation across all students using Kendall Tau, and (b) decomposes rank correlation across the first two quartiles using the *Precision@Quartile* metric. The model that combines self- and instructor-assigned scores is significantly better at predicting the top-performing students (first quartile). Combining instructor grades with self-assessment significantly improves both rank measures, especially when only a few questions are graded. Note that the total number of questions in the study was 30; we display the results up to 15, as the differences between both models is not substantial beyond that.

where  $\mathbf{s}$  and  $\hat{\mathbf{s}}$  are the true and inferred student ability parameters, respectively, and  $N_{\text{pairs}}^{\text{correct}}$  and  $N_{\text{pairs}}^{\text{wrong}}$  is the number of student pairs that are ordered correctly in the inferred ranking (with respect to the true ranking) and the number of pairs that are ordered incorrectly, respectively. Kendall Tau is equal to  $+1$  when the rankings are consistent and to  $-1$  when the rankings are inverted. The corresponding results are shown in Figure 7.2.

Three models were evaluated:

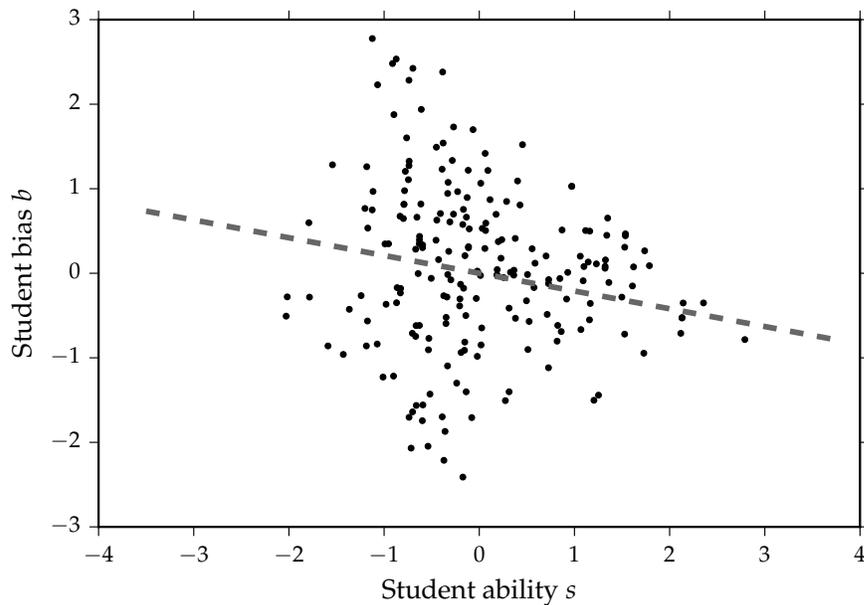


Figure 7.5: Bias vs. ability (centered). Both parameters were inferred using all of the available data. Each point in the scatter-plot corresponds to one student. A weak, but significant correlation between bias and ability exists.

- **Self-grading only:** Only students' self-submitted scores  $\log(\theta_{ij})$  are used in fitting the Rasch model parameters. All students submit their self-scores for all questions. The correctness of students' responses is not used in fitting the Rasch parameters.
- **Instructor-grading only:** Only the correctness of the responses is used for fitting the Rasch model parameters; this is a classic Rasch model. We vary the number of questions used in fitting the model parameters ( $x$ -axis in Figure 7.2).
- **Self-grading + instructor-grading:** A combination of self-scores submitted by all students for all questions and the correctness of a subset of submitted questions is used for fitting the Rasch model parameters (number of questions used is the  $x$ -axis in Figure 7.2).

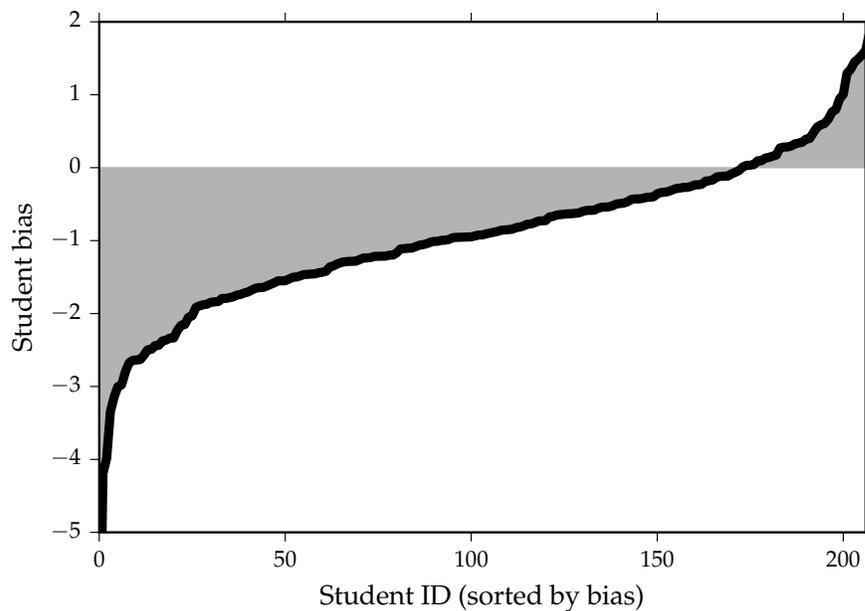


Figure 7.6: Inferred bias parameter of each student (sorted in an increasing order). The bias parameter was inferred using all of the available data.

In the case where the students in the class are relatively unbiased (low  $\sigma_b^2$ ) (top left in Figure 7.2), self-scoring achieves a better rank-correlation than the traditional IRT Rasch model, even when many questions are instructor-scored. Interestingly, in the regime of low bias, including actual instructor-graded responses actually negatively affects the correlation (this is due to over-fitting caused by a small number of instructor grades—introducing additional bias variables requires a sufficient number of observations to infer them reliably; this performance drop eventually disappears when a sufficient number of questions is included). As the bias of the population increases, the performance of the self-scoring model decreases but still exceeds the performance of the instructor-only Rasch IRT, especially in situations where only a few questions are scored.

## 7.4.2 User study

To evaluate the efficacy of the proposed self-grading approach, we conducted a user-study on Amazon’s Mechanical Turk. We solicited 206 subjects to participate in a task titled “Do a short math quiz and earn bonus!”. The subjects were asked to answer 30 math questions of varying difficulty levels ranging from basic arithmetic to pre-calculus. The questions from the dataset introduced by [104] were used in our experiment. All questions were multiple choice and included a “none of the above” option, included in order to minimize the probability of getting a right answer through a process of elimination. Although in practice, multiple-choice questions mostly defeat the purpose of self-grading, we use multiple choice questions for the ease of evaluation and the lack of subjectivity that would be otherwise present in free-response questions. Figure 7.3 illustrates a single question from the task. The subjects were asked to mark what they believed to be the correct answer, and then to assign themselves the number of points that they would receive if they answered the question correctly. The input was provided through a slider. Moving the slider automatically displayed the number of points that the subject would gain if they answered the question correctly (green), and the number of points they would lose if they answered the question incorrectly (red). The points were then converted to currency (1 point = \$0.01), and paid through a “bonus” mechanism in Mechanical Turk. We chose to use real currency as a reward to ensure that the subjects had a stake in their performance, and thus there is incentive to think carefully about their self-assigned scores.

We follow the same evaluation scheme that we described in the previous section. Recall, that we are interested in the quality of the assessment derived

from the students' self-evaluation. In the simulation study, a "gold-standard" assessment was available and allowed us to use rank correlation between the "gold-standard" ranking and the inferred ranking as an evaluation metric. In this user-study, we consider the ranking inferred by the IRT model that relies on the complete dataset, as a proxy for the "gold-standard" ranking. We then repeat the evaluation scheme described in the previous section: (i) vary the number of instructor-graded questions from 0 to all questions (30) and combine that with the self-assigned scores for every question, (ii) infer the ranking using the proposed model, and (iii) compare it to the ranking that is derived from "gold-standard" proxy.

We find that the results are comparable to those obtained in the simulation (Figure 7.3.3(a)). Self-scoring is already able to obtain a reasonable correlation with the "gold-standard" ranking even without any instructor-graded question. Incorporating instructor-grades for additional questions improves the performance. Rank correlation metrics, such as Kendall Tau, while convenient for summarizing the results with a single quantity, often fail to distinguish regimes where the model might perform differently. It is instructive to consider the performance of rank-correlation in the different segments of the ranking. Figure 7.3.3(b) decomposes the results by quartiles. We employ a more intuitive metric, *Precision@Quartile*, defined as follows:

$$Precision@Q_i = \frac{|\hat{S}_{Q_i} \cap S_{Q_i}|}{|\hat{S}_{Q_i}|}$$

where  $S_{Q_i}$  is the set of students in the  $i$ th quartile of the "gold-standard" ranking, and  $\hat{S}_{Q_i}$  is the set of students in the  $i$ th quartile of the inferred ranking. This metric captures the ability of the model to perform within a particular segment of the ranking. For example, looking at Precision at the first quartile, measures the ability of the model to predict top students. From Figure 7.3.3(b) we can conclude

that the model is significantly better at distinguishing the top-ranked students (first quartile) as compared to the lower-ranked students (second quartile). By using the self-scoring signal without any instructor-graded questions, we are able to recover nearly 60% of the top quarter of all students. The performance in the second quartile is significantly lower, but follows the same trend: incorporating the students' self-reported scores in the regime of zero to several questions significantly improves performance over the baseline of instructor-graded questions alone. This observation leads to the conclusion that, at least in this study, better students were better at estimating their ability. We look into the effect of self-estimation performance in more detail in the next section.

### 7.4.3 Self-assessment and bias

The performance of the model that relies on self-assessment depends fundamentally on the model's estimates of the students' biases as well as the ability of the students to self-assess reliably (self-assessment variance). In our model, we infer only the individuals' biases and assume constant variance in the self-assessment likelihood (these could in principle be estimated as well. Figure 7.6 illustrates the individual inferred biases for each student (averaged across multiple folds), sorted in an increasing order. The resulting distribution illustrates the skew in the bias distribution towards "under-confidence," i.e., most students tend to under-estimate their ability (act conservatively). The importance of estimating bias is underlined in Figure 7.3.3(a), where we include an additional baseline **Self-Scored + Graded (no bias)** (light solid line). This baseline combines self-assessment and instructor-grades but does *not* incorporate the explicit student-bias parameter. As evident from the graph, estimating bias is critical for

combining self-grading and instructor-grading: without the bias parameter, the model is not able to leverage the benefits of both signals.

It is potentially insightful to investigate the relationship between self-assessment bias and ability. We consider the inferred bias parameter after incorporating instructor-grades for all questions, and compare it to the inferred ability parameter of each student. The result is illustrated in the scatter-plot in Figure 7.5. While the relationship between the two is not strong, there exists a negative correlation between ability and self-assessment bias (Pearson's correlation: 0.17,  $p$ -value = 0.013). Students that are more able tend to underestimate their ability, and students that are less able tend to inflate their ability. This finding is consistent with the literature in self-assessment [188, 182].

## 7.5 Conclusion and Future Work

In this chapter, we have developed a novel approach for performing calibrated, summative self-assessment by combining (i) student's self-evaluations obtained via an incentive-compatible scoring mechanism and (ii) a minimal number of instructor-graded responses. We have shown that when the scoring rule is quadratic, the standard IRT Rasch model reduces to standard linear regression. We have demonstrated that the quality of the inferred assessment using self-scoring alone without additional instructor input is, on-average, comparable to the performance obtained using the standard IRT that requires significant instructor effort. Furthermore, by incorporating a minimum number of instructor-graded responses, we have shown that our approach substantially improves the estimates of the students' abilities and the questions' difficulties. Finally, we

have addressed the long-standing issue of applying scoring rules in practice: dealing with the consequences of individuals' biases and non-risk-neutrality. We have proposed to explicitly model the combined effect of these two factors within the standard IRT framework, allowing the model to effectively de-bias these individual differences.

## CHAPTER 8

### LEARNING VOCABULARY FROM READING THE WEB

#### 8.1 Introduction

In this, and the following two chapters, we shift focus from assessment, and towards learning content curation – the second component of mastery learning. In this chapter, we explore the potential of the web to be utilized as a resource for teaching foreign language vocabulary. As more and more people shift to reading electronic media (e.g., Wikipedia, news articles, e-books), we are presented with an opportunity to exploit that content as a vehicle for teaching a foreign language. Specifically, in this chapter, we explore the potential of teaching readers vocabulary in a new language by deliberately replacing words in the reading with their translation in the foreign language. Our hypothesis is that the reader will implicitly pick-up the meaning of the foreign words from context, while reading content that they are interested in and understand. We develop a computational approach for modeling word learning from context, and utilize this model to optimize a set of web pages to present to the reader to maximize the number of acquired words.

Today, an adult trying to learn a new language is likely to embrace an age-old and widely accepted practice of learning vocabulary through curated word lists and rote memorization. Yet, it is not uncommon to find yourself surrounded by speakers of a foreign language and instinctively pick up words and phrases without ever seeing the definition in your native tongue. Hearing “*pass le sale please*” at the dinner table from your in-laws visiting from abroad, is unlikely to make you think twice about passing the salt. Humans are extraordinarily

good at inferring meaning from context, whether this context is your physical surrounding, or the surrounding text in the paragraph of the word that you don't yet understand.

This phenomenon is known as *incidental vocabulary acquisition*, and traces an extensive body of research exploring the necessary conditions for acquiring words through reading. The key findings of research into L2 (second language) incidental vocabulary learning is that the degree to which a word is acquired from reading depends on three factors: (i) the number of exposures to the word, (ii) the contexts in which the word appears and (iii) the learner's attention during reading. In this chapter, we explore the possibility of deliberately introducing learners to foreign words by deliberately "switching" some of the words in a reading (written in the reader's native language) to their translations in the foreign language. Incidentally, this phenomenon occurs "in the wild" and is termed *code-switching* or *code-mixing*, and refers to the linguistic pattern of bilingual speakers swapping words and phrases between two languages during speech. While this phenomenon had received significant attention from both a socio-linguistic [126] and theoretical linguistic perspectives [13, 15] (including some computational studies), only recently has it been hypothesized that "code-switching" is a marking of bilingual proficiency, rather than deficiency [59].

In this chapter, we develop an approach for automatically generating such "code-switched" text with an explicit goal of maximizing the lexical acquisition rate. Our method is based on the principle of maximizing the number of exposures to the word, and involves (i) finding a minimal set of documents (e.g., webpages) to present to the learner and (ii) the words to "switch" in each

document, such as to optimize the expected number of acquired words. The contributions of the work described in this chapter are:

- Based on the user studies we conduct, we develop a model that relates the number of exposures to a word to the likelihood of that word's retention.
- Based on another user study we conduct, we study the effect of the number of "switched" words within a context to the likelihood of the words' retention.
- Based on these findings, we develop an optimization objective for (i) selecting documents to present to the reader and (ii) the words to "switch" in each document, in order to maximize the expected number of words recalled.
- We conduct an evaluation of the proposed algorithm via a user study and demonstrate its effectiveness and limitations in a subset of word frequency regimes.

## 8.2 Related Work

Our proposed approach to the computational generation of code-switched text, for the purpose of L2 pedagogy, is influenced by a number of fields that studied aspects of this phenomenon from distinct perspectives. In this section, we briefly describe a motivation from the areas of socio- and psycho- linguistics and language pedagogy research that indicate the promise of this approach.

### 8.2.1 Incidental vocabulary acquisition

Vocabulary acquisition through reading had received significant attention from psycholinguistics and education researchers. While there is a consensus among these fields that a significant portion of vocabulary acquisition occurs through extensive reading, the exact mechanisms for what contributes to effective word learning from reading is not well understood [81]. Three factors are often believed to play a role in incidental vocabulary acquisition from reading: (i) the number of exposures to the word, (ii) the contexts in which the word appears and (iii) the attention and cognitive effort of the reader. Among them, frequency of the word (number of exposures) is widely agreed to be the most important predictor of vocabulary acquisition [78, 79, 209]. The effect of context in which the word appears, however, is disputed. For example [169] argue that contexts that illuminate the meaning of the word are critical in facilitating the acquisition of the word's meaning, while [131, 139] argue that contexts where the word's meaning is obvious are conducive of reading rather than word learning [209]. Recent neurophysiological findings, however, described in more detail later in this section [21] support the hypothesis that contexts may play a role in word acquisition. There is, however, a consensus in the literature, that the learner must comprehend most of the words (more than 80%) in the context to be able to acquire the meaning of new words [73, 74, 169, 77, 105].

In this chapter, we develop a formal optimization objective for generating mixed-language texts (code-switched text) based on the following well-accepted findings in L2 incidental word learning literature: (i) frequency of the word is the most important predictor of that word's retention and (ii) the majority of the context must be comprehensible for the learner to create an opportunity for

word acquisition from context.

### **8.2.2 Code-switching as a natural phenomenon**

Code-switching (or code-mixing) is a widely studied phenomenon that received significant attention over the course of the last three decades, across the disciplines of sociolinguistics, theoretical and psycholinguistics and even literary and cultural studies (predominantly in the domain of *Spanish-English* code-switching) [111].

Code-switching that occurs naturally in bilingual populations, and especially in children, has for a long time been considered a marking of incompetency in the second language. A more recent view on this phenomenon, however, suggests that due to the underlying syntactic complexity of code-switching, code-switching is actually a marking of bilingual fluency [59]. More recently, the idea of employing code-switching in the classroom, in a form of conversation-based exercises, has attracted the attention of multiple researchers and educators [132, 116], yielding promising results in an elementary school study in South-Africa.

### **8.2.3 Computational approaches to code-switching**

Additionally, there has been a limited number of studies of the computational approaches to code-switching, and in particular code-switched text generation. Solorio and Liu [181], record and transcribe a corpus of Spanish-English code-mixed conversation to train a generative model (Naive Bayes) for the task of pre-

dicting code-switch points in conversation. Additionally they test their trained model in its ability to generate code-switched text with convincing results. Building on their work, [1] employ additional features and a recurrent network language model for modeling code-switching in conversational speech. Adel and colleagues [18] propose a statistical machine translation-based approach for generating code-switched text. We note, however, that the primary goal of these methods is in the faithful modeling of the natural phenomenon of code-switching in bilingual populations, and not as a tool for language teaching. While useful in generating coherent, syntactically constrained code-switched texts in its own right, none of these methods explicitly consider code-switching as a vehicle for teaching language, and thus do not take on an optimization-based view with an objective of improving lexical acquisition through the reading of the generated text. The idea of exploiting code-switching with extensive reading on the web to facilitate vocabulary acquisition was explored in [190] and more recently implemented in Google’s Language Immersion app. The work of [190] demonstrates that learners are able to learn new words when exposed to them in the context of reading, however, they do not address the task of how to select the documents or the words to “switch”. In this work, we focus explicitly on the problem of optimizing content presentation to facilitate vocabulary acquisition.

#### **8.2.4 Computational approaches to sentence simplification**

Although not explicitly for teaching language, computational approaches that facilitate accessibility to texts that might otherwise be too difficult for its readers, either due to physical or learning disabilities, or language barriers, are relevant. In the recent work of [88], for example demonstrates an approach to increasing

readability of texts by learning from unsimplified texts. Approaches in this area span methods for simplifying lexis [206, 17], syntax [177, 178], discourse properties [83], and making technical terminology more accessible to non-experts [48]. While the resulting texts are of great potential aid to language learners and may implicitly improve upon a reader's language proficiency, they do not explicitly attempt to promote learning as an objective in generating the simplified text.

### **8.2.5 Recent neurophysiological findings**

Evidence for the potential effectiveness of code-switching for language acquisition, stem from the recent findings of [21], who have shown that even a single exposure to a novel word in a constrained context, results in the integration of the word within your existing semantic base, as indicated by a change in the N400 electrophysiological response recorded from the subjects' scalps. N400 ERP marker has been found to correlate with the semantic "expectedness" of a word [96], and is believed to be an early indicator of word learning. Furthermore, recent work of [57], show that word surprisal predicts N400, providing concrete motivation for artificial manipulation of text to explicitly elicit word learning through natural reading, directly motivating our approach. Prior to the above findings, it was widely believed that for evoking "incidental" word learning through reading alone, the word must appear with sufficiently high frequency within the text, such as to elicit the "noticing" effect — a prerequisite to lexical acquisition [167, 36].

### 8.3 Model (part 1)

In formulating the model, we are going to assume that we have access to a pool of documents  $\mathcal{D} = \{D_k\}$ , where each document is a set of contexts  $D_k = \{C_i^{(k)}\}$  (e.g., a sentence), and each context is a set of words:  $C_i^{(k)} = \{w_{ij}\}$ . Consider that each word  $w_{ij} \in C_i^{(k)}$  can exist in one of two states, represented by  $x_{ij}^{(k)} \in \{0, 1\}$ , where  $x_{ij}^{(k)} = 0$  indicates a word that remains in the reader's native language, while  $x_{ij}^{(k)} = 1$  represents a word that is "switched" to its translation in a foreign language. Let  $X_i^{(k)} = \{x_{ij}^{(k)}\}$  be a set of such variables indicating the state of each word in context  $i$  of document  $k$ . Finally, let  $X^{(w)} = \{x_{ij}^{(k)} \mid w_{ij} = w \wedge w_{ij} \in C_i^{(k)} \in D_k \in \mathcal{D} \forall i, j, k\}$ , i.e., all state variables across all contexts and documents that correspond to word  $w$  (e.g.,  $w = \text{"cat"}$ ).

Using this formalism, we can now express the total number exposures of the reader to the translation of any given word, given the state of that word across all documents and their contexts. Let  $n_w$  be the total number of exposures of the reader to the translation of word  $w$ , which can be expressed as:

$$n_w = \sum_{x_{ij} \in X^{(w)}} x_{ij} \quad (8.1)$$

We are interested in the probability that the reader recalls word  $w$  after  $n_w$  exposures. We are thus interested in the mapping of  $n_w$  to a probability  $P(\text{recall } w \mid n_w)$ :

$$P(\text{recall } w \mid n_w) = f(n_w) \quad (8.2)$$

where  $f$  is a function that maps the number of exposures to the translation of the word to the likelihood the reader recalls the word. In the next section, we will perform an empirical study to validate a good choice for  $f$ . Intuitively, we expect  $f$  to be an increasing function, capturing the consensus in literature and

the intuition that the number of exposures to the word increases the likelihood that the reader learns the word.

Taking into account all words that are mentioned in the document set  $\mathcal{D}$ , which we denote with set  $W(\mathcal{D})$ , the expected fraction of the words recalled by the reader after reading all documents in  $\mathcal{D}$  is given by:

$$\mathbb{E}[\text{number of words recalled after reading } \mathcal{D}] = \sum_{w \in W(\mathcal{D})} P(\text{recall } w \mid n_w)$$

This allows us to formulate our optimization objective as maximizing the expected number of acquired words:

$$\operatorname{argmax}_{X^{(w)} \forall w \in W(\mathcal{D})} \sum_{w \in W(\mathcal{D})} P(\text{recall } w \mid n_w) \quad (8.3)$$

where we are optimizing over the state of the switching variables  $X^{(w)}$  for every word that appears in the document set  $\mathcal{D}$  (recall that the switching variables determine  $n_w$ , on which the probability of recall of word  $w$  depends). Figure 8.1 illustrates the optimization problem. In the following section, we conduct two studies to (i) evaluate a suitable functional form for modelling  $P(\text{recall } w \mid n_w)$  and (ii) observe the effect of the number of switched words in a context on recall. We return to completing the formulation of the optimization problem in Section 8.5.

## 8.4 User studies

### 8.4.1 Corpus

We conduct a study on Amazon Mechanical Turk (AMT) with a goal of observing the relationship between the number of exposures to the translated word and

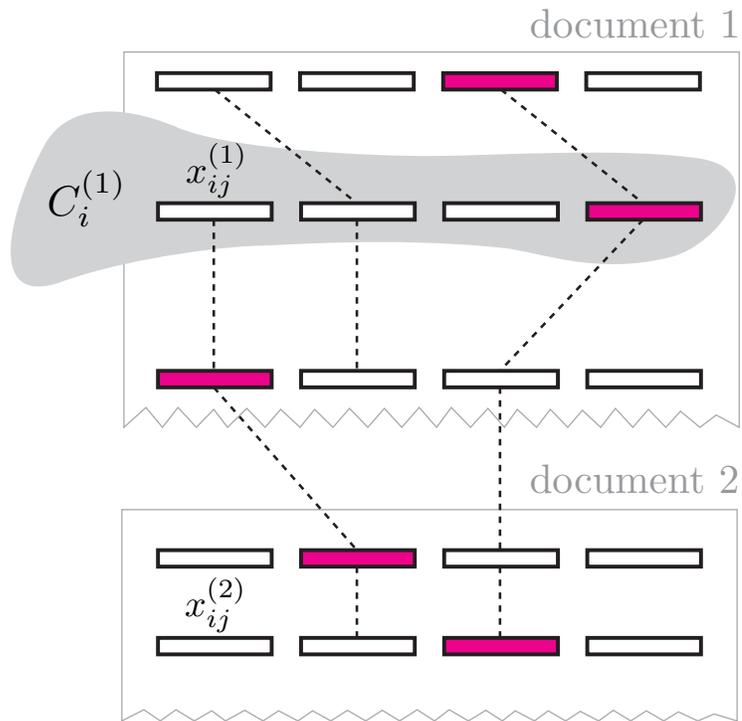


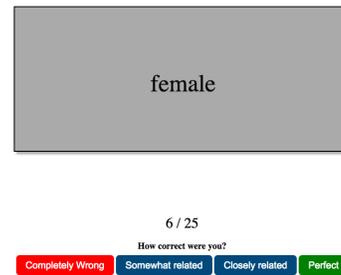
Figure 8.1: Schematic of the optimization problem for selecting words within documents to “switch”, i.e., replace their occurrence with their translation into a foreign language. In this example, pink words represent words that are “switched” ( $x_{ij} = 1$ ), while white words remain in their original language ( $x_{ij} = 0$ ), and dashed lines trace the mentions of the same word within and across documents. A gray shaded region represents a context, e.g., a sentence.

the likelihood of its recall following the reading. We collect a dataset of 846 Wikipedia articles (with a minimum of 250 and a maximum of 3000 words) from the category of Unusual articles, which contain articles that “... are a bit odd, whimsical, or something you would not expect to find in Encyclopedia Britannica”<sup>1</sup>, with the goal of attracting the reader’s attention and focus, as shown to be important in the process of incidental vocabulary acquisition [49, 156, 166]. Within each article, we consider a context to be a single sentence, and we consider

<sup>1</sup>[https://en.wikipedia.org/wiki/Wikipedia:Unusual\\_articles](https://en.wikipedia.org/wiki/Wikipedia:Unusual_articles)

In humans and **SEIJKA** complex life forms , blood and lymph circulate in two different systems , the circulatory system and lymphatic system , which are enclosed by systems of capillaries , veins , arteries , and nodes . This is known as a closed circulatory **SHOUBOUSHI** . Insects , however , have an open circulatory **SHOUBOUSHI** in which blood and lymph circulate unenclosed , and mix to form a substance called hemolymph . All organs of the insect are bathed in hemolymph , which provides oxygen and nutrients to all of the insect 's organs .

(a) *reading stage*



(b) *quiz stage*

Figure 8.2: Screenshots of the two stages of the Mechanical Turk studies described in this chapter. The *reading stage* (a) presents readers with an article in English, with a subset of the words switched to a foreign language. The follow-up *quiz stage* presents the readers with a subset of the words they encountered during the reading, one at a time in a random order, requesting the worker to enter their guess and evaluate the precision of their guess after being revealed the meaning of the word.

a subset of the words in each sentence as potential candidates for “switching” (i.e., translating into a foreign language). The subset of words that is considered as candidates is selected from a finite vocabulary that we prepare based on this corpus. Specifically, we select only nouns and verbs that appear at least 4 times across all documents in the corpus. In the next two sections, we describe the user studies based on this corpus in detail.

## 8.4.2 Study I: Modelling word acquisition rate

In order to study the effect of the number of occurrences of the switched word (from hereon we refer to as *word frequency* or *item frequency*) on retention, we conduct an experiment on 4 articles selected from the corpus described above. In each article, we switch at most 1 word per context, determining the switched

word by giving priority to the most-frequent word in the article, and breaking ties randomly. We perform this “switching” procedure twice for each of the 4 articles, randomizing the foreign vocabulary within each pair, yielding a total of 8 mixed-language articles whose switched words exhibit a wide distribution over word frequency. A total of 135 workers were solicited for a task titled “Read an article with a twist” and were instructed to read the article completely and told about a follow up quiz. The article was presented in English, with the exception of switched words, which were replaced with Japanese words<sup>2</sup>. After completing the reading, the workers were given a vocabulary quiz, preceded by a tutorial explaining the quiz procedure. The vocabulary quiz was self-graded – workers were presented with flashcards, one at a time, and a question asking whether they think they know the meaning of the word. If the worker answered “yes”, the flashcard was “flipped” revealing the English meaning, and a set of 4 options to select from, indicating how close their guess came to the true meaning. Table 8.1 summarizes the criterion for choosing each option, presented in an expanded form to the workers during a tutorial that proceeded the quiz. The workers were required to type in their guess and requested to be thoughtful in evaluating their input. Fifteen of the most frequent words in each of the four articles were presented in the quiz. Figure 8.4.2 displays the screenshots of the reading and the quiz stages of the Mechanical Turk task. Words at frequency rank<sup>3</sup> 10 and lower appeared at most twice in any of these articles, and thus including additional words at lower ranks would likely not be informative.

---

<sup>2</sup>[https://en.wiktionary.org/wiki/Appendix:1000\\_Japanese\\_basic\\_words](https://en.wiktionary.org/wiki/Appendix:1000_Japanese_basic_words). Note that the Japanese words were not actual translations and were assigned randomly as translations to the words in English. The workers were post-tested to ensure that none were familiar with Japanese. The reason for randomizing word assignments instead of using real translations is to ensure consistent vocabulary across different articles to reduce variance in the results from the differences in the words

<sup>3</sup>recall again that we use the term word frequency to refer to the number of occurrences of the word when it is “switched”

We present our results by averaging workers' scores for words binned by their frequency in their corresponding documents. In order to gain understanding into the effect of frequency on the strength of word acquisition, we display the results for each of the four levels of the score, which we define as  $recall@k$ . We define  $recall@k$  as follows:

$$recall@k = \begin{cases} 1 & \text{if } score > k \\ 0 & \text{otherwise} \end{cases} \quad (8.4)$$

where  $k \in \{0, 1, 2, 3\}$  is the degree to which the reader acquired the meaning of the word (i.e., level of recall or strength of recall). For example  $recall@0 = 1$  means that the reader thought they knew the word, but did not correctly identify its meaning, while  $recall@0 = 0$  means that the reader did not recognize the word. Similarly  $recall@3 = 1$  means that the reader correctly identified the meaning of the word (see Table 8.1 for more details on each level). The results are shown in Figure 8.3, with each curve displaying the average  $recall@k$  (averaged across all workers) for different levels of recall  $k$ , as a function of word frequency. Across all recall levels, it is evident that words with higher frequency (referred to as item frequency in the plot) are more likely to be recalled. There is, however, an important difference in the relationship between frequency and recall at low and high recall levels. For all recall levels  $> 0$ , there is a clear frequency threshold, below which recall is not affected significantly by an increase in frequency. Recall at level  $> 0$  (i.e., when the reader believes that he or she have learned the word, but did not correctly identify its meaning), however, increases monotonically with frequency. All recall levels, however, show an effect of saturation, i.e., exposure to the word more than a certain number of times (between 30 and 40 in Figure 8.3) does not improve recall (i.e., diminishing returns). Note, however, that the study is performed on a short time-scale, i.e., not considering the effect of

Recall level $k$	Option	Description
1	Completely wrong	Your guess was not close at all. For example, if the meaning of the word was "DOG" and you guessed "TABLE", then you are completely wrong.
2	Somewhat related	Your guess was not entirely unrelated. An example of this is if the real meaning was "EATING" and you guessed "WATCHING". Although the meaning of the words is different, they are both actions that a person can perform. This means that you were able to extract some degree of understanding about the word from the reading.
3	Closely related	The words are closely related. If for example the real meaning was "EATING" and you guessed "DRINKING", or if the real meaning was "DOG" and you guessed "CAT" or "ANIMAL". The words may not mean the same thing, but they are closely related.
4	Perfect	You correctly guessed the meaning of the word. Note that it's OK if your guess was a slight variation on the correct meaning. For example if the revealed word was "EAT" and you wrote "EATING" or if the revealed word was "DOG" and you wrote "PUPPY" – all of these are considered Perfect.

Table 8.1: A categorization shown to the worker during the tutorial to guide in evaluating the quality of their guess.

repetition beyond a single reading session. Our interest in this work is to model the relationship between frequency and recall to use it in optimizing a presented set of documents. Longer-term, longitudinal studies on recall are likely to inform a model of higher fidelity, however, the principle of optimizing it is likely to be the same. In Section 8.5, we develop a model that captures the behavior observed in Figure 8.3.

### 8.4.3 Study II: Modelling effect of context

In this section, we study the effect of context on recall. Specifically, we are interested in understanding how the number of words "switched" within a

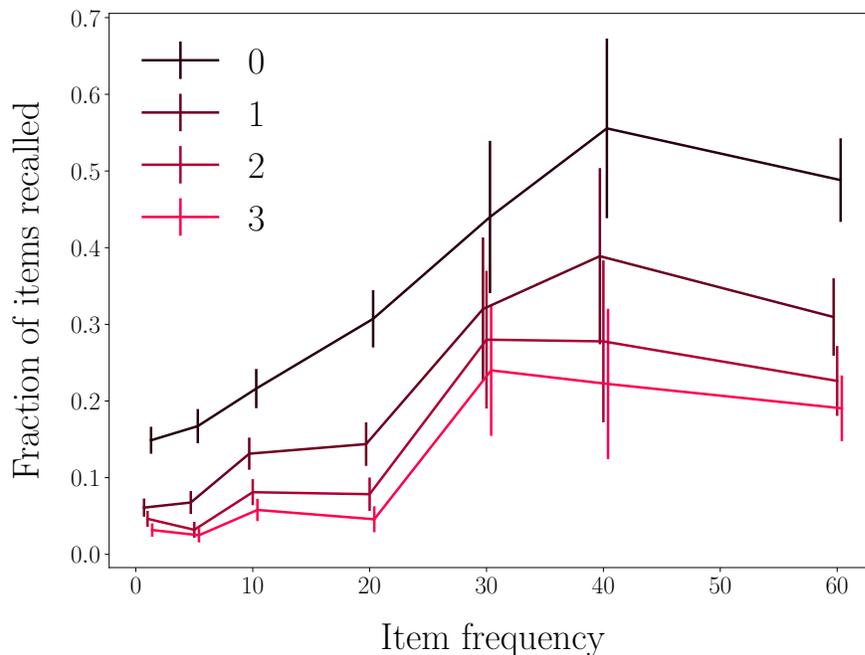


Figure 8.3: Relationship between word (item) frequency and recall at different recall levels (higher recall levels correspond to more precise recall of the word’s meaning). Item frequency refers to the number of occurrences of the word switched to its translation in the document where it occurs.

single context (sentence in our experiments) affects recall. Intuitively, we expect a trade-off: “switching” more words to their translation within a single context creates more learning opportunities, but also increasingly obscures the context, potentially reducing the effectiveness of the additional encounters with the word. We conduct an additional study on Amazon Mechanical Turk, with the goal of understanding how the number of “switched” words within a single context affects recall. We use a single article, but generate four different mixed-language versions (experimental conditions), progressively increasing the number of “switched” words in a single context. Within each context, we select the top  $k$  most frequent words in the document, and break ties randomly<sup>4</sup>. As

<sup>4</sup>although not every context may have exactly  $k$  candidates (corresponding to the condition) for switching, the vast majority do, as we confirm by verifying the median number of switched

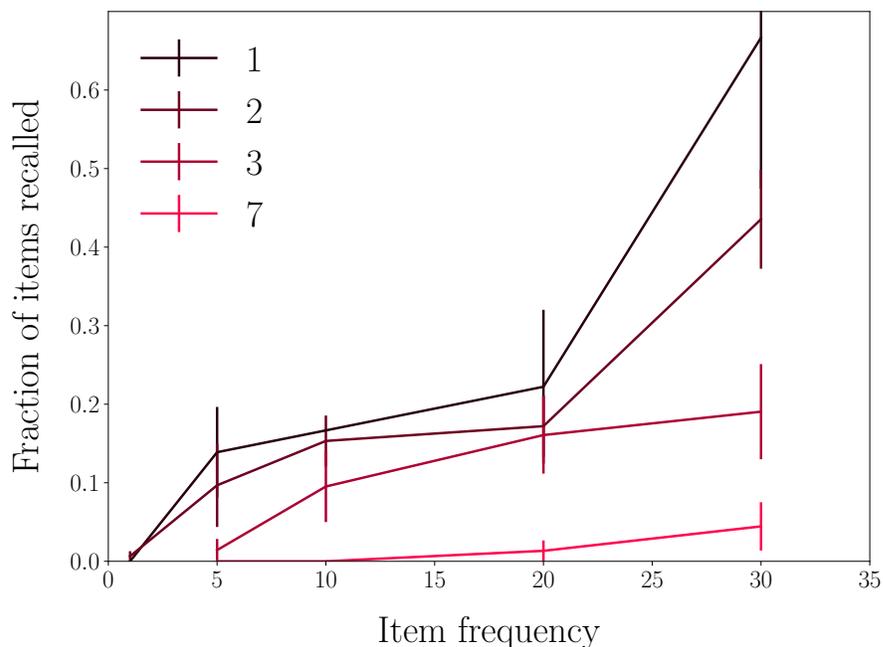


Figure 8.4: Relationship between word (item) frequency and recall for different number of “switched” words per context. Item frequency refers to the number of occurrences of the word switched to its translation in the document where it occurs.

in the first study described in the previous section, we quiz the workers only on the 15 most frequent words. This introduces a subtlety into our evaluation that requires some elaboration. Increasing the number of “switched” words per context, of course, increases the total number of words that the reader was exposed to during reading. In this sense, it may appear unfair to evaluate all conditions on the same number of words (15), which we do in order to balance the workload across all conditions. Figure 8.4, which summarizes the result for this experiment, however, explains why this does not affect the conclusion that we can draw based on the results. Figure 8.4, similar to Figure 8.3, presents recall as a function of word frequency, but where each curve corresponds to the number of “switched” words per context (experimental condition). All curves are drawn

---

words in the document

at the recall level of 1 (*recall@1*), though the curves for the other levels exhibit a similar characteristic. We make two important observations based on Figure 8.4: (i) increasing the number of “switched” words per context significantly reduces recall across all word frequencies and (ii) the lowest word frequency is between 5 and 10 for the two conditions corresponding to 3 and 7 “switched” words per context (i.e., the two densest conditions), and the recall for those words is already nearly zero. This indicates that although there are more “switched” words to which the reader was exposed in conditions 3 and 7, those words would be of even lower frequency, for which the recall would also be likely approximately zero. This leads to an important conclusion: although “switching” more words per context does expose the reader to more words, it does not compensate for the significant drop in effectiveness of those contexts which leads to a significant drop in recall for words across all frequencies in the text. The optimal number of “switched” words per context (sentence) based on our results is one. In the following section, we return to the formulation of the optimization problem for selecting documents and words to “switch”, and formally encode the findings in this and the previous sections.

## 8.5 Model (part 2)

In this section, we integrate the findings from the two studies in the previous section, to complete the development of the optimization problem that we began in Section 8.3. Based on the relationship between word frequency and recall observed in Figure 8.3, we propose a simplified model illustrated in Figure 8.5 that captures the key observations: (i) multiple occurrences of the “switched” word below a certain frequency threshold do not have a significant affect on

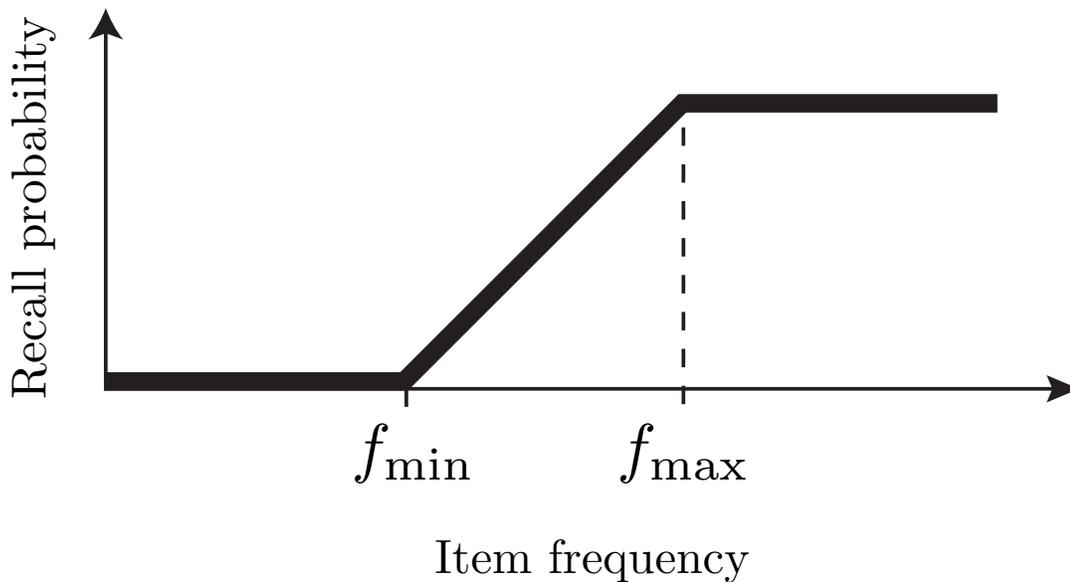


Figure 8.5: Proposed model for the relationship between word frequency and recall, based on the empirical findings in Figure 8.3.

recall, (ii) above the “minimum frequency threshold”, the relationship between frequency and recall is approximately linear and (iii) above a certain “maximum frequency”, increasing the frequency of the word leads to diminishing returns, i.e., does not significantly affect recall. Figure 8.5 models these three observations with a piecewise linear function  $f(n_w)$  (that parametrizes the probability of recall in Equation 8.2), with a minimum frequency threshold  $f_{\min}$  and a maximum frequency threshold  $f_{\max}$ . Based on the empirical observations in Figure 8.3, we set  $f_{\min} = 5$  and  $f_{\max} = 30$ .

We now integrate our finding from Figure 8.4, i.e., that presenting more than one “switched” word per context significantly reduces recall in all frequency regimes. This can be naturally incorporated as a constraint into the optimization

problem given in Equation 8.5:

$$\begin{aligned} & \operatorname{argmax}_{X^{(w)} \forall w \in W(\mathcal{D})} \sum_{w \in W(\mathcal{D})} P(\text{recall } w \mid n_w) & (8.5) \\ & \text{such that } \sum_{x_{ij}^{(k)} \in X_i^{(k)}} x_{ij}^{(k)} \leq 1 \forall i, k \end{aligned}$$

ensuring that at most one word per context is “switched”. Solving this optimization problem, however, will select words from an entire collection of documents  $\mathcal{D}$ , which may consist of the entire Wikipedia article collection. Our task, however, is to present the reader with only a small subset of the collection to read on any given day. As such, we require an additional set of constraints that select a finite number of documents from the complete collection of documents. Moreover, because different documents may be of drastically different lengths, a more practical constraint may be on the total number of words or sentences that the reader is presented with, on say a particular day. Let  $S(D_k)$  be the total number of sentences in document  $D_k$ , and let  $y_k \in \{0, 1\}$  be a binary variable that indicates that document  $D_k$  is selected to be shown to the reader. The complete optimization problem can then be expressed as:

$$\begin{aligned} & \operatorname{argmax}_{\forall w \in W(\mathcal{D})} \sum_{w \in W(\mathcal{D})} P(\text{recall } w \mid n_w) \\ & \text{such that } \sum_{x_{ij}^{(k)} \in X_i^{(k)}} x_{ij}^{(k)} \leq 1 \forall i, k & (8.6) \\ & \sum_k S(D_k) y_k \leq S_{max} \forall k \\ & x_{ij}^{(k)} \leq y_k \forall i, j, k \end{aligned}$$

where  $S_{max}$  is a user-specified parameter that indicates the maximum number of sentences that he or she are willing to read at any given time (this could also be expressed in terms of the maximum number of words or documents). The

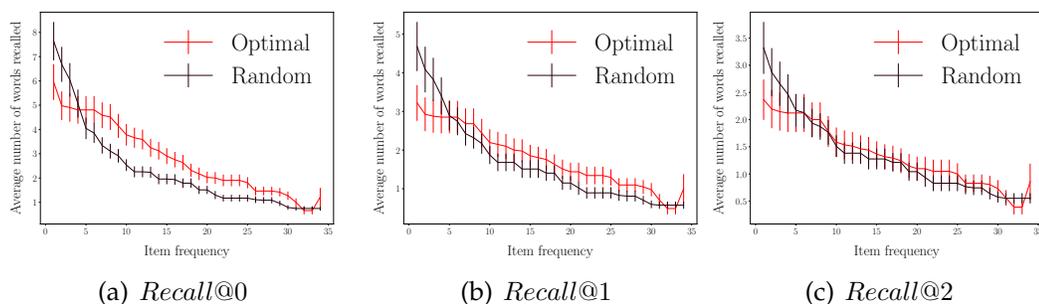


Figure 8.6: Average number of words recalled as a function of word (item) frequency, partitioned by recall level and condition.

last constraint in the above optimization problem ensures that a document is shown to the user ( $y_k = 1$ ) if at least one word in it is “switched”. The above optimization problem (with a piecewise linear objective in Figure 8.5) can be reformulated as an Integer linear program (ILP), which we solve using the Gurobi solver<sup>5</sup>.

## 8.6 Experiments

We conduct a user study to evaluate the effectiveness of the proposed model and the optimization problem for presenting the articles and selecting words to “switch” to their translations. Our user study consists of presenting a user with three articles obtained in one of two ways: (i) via solving the optimization problem in Equation 8.6 (**Opt** condition) or (ii) randomly (**Rand** condition). The **Opt** condition articles are obtained as follows: we randomly partition our Wikipedia corpus described in Section 8.4.1 into five folds, each with approximately 200 articles. We solve an optimization problem in Equation 8.6 with the constraint on the maximum number of sentences (500) and the maximum number of documents

<sup>5</sup><http://www.gurobi.com/>

(3) over the documents in each fold. The resulting articles are presented to the readers in the **Opt** condition. The **Rand** condition articles are obtained as follows: five folds of three articles each are drawn from a subset of our Wikipedia corpus at random but with a constraint to ensure that the average length of each triplet of articles is the same as the length of articles from the **Opt** condition (to ensure that the observed differences between conditions are not due to the difference in the articles' length). Within each sentence of each article, the most frequent word in that document is "switched", with ties broken randomly. Selecting the most frequent word in each context results in the strongest possible random baseline, as higher item frequency promotes better recall (Figure 8.3). As in the previous experiments, the users were tested on the 15 most frequent words across the three documents that they were presented with. Also the words in the same frequency rank (e.g., all most frequent words or all second most frequent words) across all conditions and documents mapped to the same foreign words (again chosen at random from Japanese). This was done to reduce variance across conditions arising as a result of the inherent differences in memorability of the individual foreign words. A total of 88 Mechanical Turk workers were solicited for the task, and randomly assigned to one of the two conditions. A random triplet of articles was then shown to each worker from their assigned condition.

We summarize the results in Figure 8.6 with an average number of recalled words of a given frequency, partitioned by recall level and condition. There are several observations that we make based on these results: (i) optimal content generation (i.e., **Opt** — selecting documents to display and words to "switch") dominates in recall across a wide range of frequencies for the recall levels 0 and 1. We also observe that recall for words of frequency 5 and lower is higher in the **Rand** condition (note the cross-over in each of the three panels of Figure 8.6).

As a consequence, comparing the average total number of words recalled across recall levels, however, does not yield a statistically significant result in difference (though the differences in the first panel for frequencies  $\geq 5$  are significant). The direct explanation for this observation is the following: because the piecewise linear objective (Figure 8.5) contains a “deadzone” for frequencies of 5 and lower (i.e., the model assigns 0 recall likelihood to words that occur 5 times or fewer), the model does not favor “switching” words that appear 5 times or fewer. More interesting, however, is the observation that exposures to a word fewer than 5 times provide noticeable gains in recall (in contrast to the assumption of our model in Figure 8.5). While  $recall@0$  was the only recall level that did not appear to feature a low frequency threshold  $f_{\min}$  (see Figure 8.3), other recall levels do in fact feature small gains from even a few exposures to a “switched” word. Our experimental results in Figure 8.6 therefore suggest that capturing those small gains is important in encouraging the model to select low-frequency words to boost recall in a low frequency regime. A natural extension to the model in Figure 8.5 is to model the low frequency regime  $freq \in [0, f_{\min}]$  with a linear relationship (but with a shallower slope than the  $freq \in [f_{\min}, f_{\max}]$  regime).

More importantly, however, the above results indicate that the optimized document set does provide a small, but measurable boost in recall, in agreement with the behavior encoded in the model, and suggests a fruitful direction of research into developing higher fidelity models for greater gains in recall across all frequency regimes. It is also expected that the optimization problem that has access to a pool with more than 200 documents (as is the case in our experiments) will deliver proportionally greater gains with access to more words and contexts. We discuss future work and extensions to our approach in the following section.

## 8.7 Conclusion and Future work

In this chapter, we have demonstrated the potential of utilizing the web for generating learning content that aids readers in incidentally acquiring foreign language vocabulary. Our key contribution is a model and an optimization problem for selecting documents and generating mixed-language text within those documents that maximizes word retention. Our work is motivated by a significant body of research in first and second language learning that identifies the number of exposures to a word as the most important factor in predicting that word's retention. Naturally, the vastness of the web (e.g., news, encyclopedia articles) provides a great opportunity for exposing readers to words in a deliberate manner that helps learners gain the most from reading. We believe that the work presented in this chapter is an important step in developing tools for optimizing content presentation on the web to facilitate vocabulary instruction. Our work also naturally opens several natural directions of inquiry that we describe in some detail below:

- **Learner model** In presenting mixed-language content to the learner, it is important to also keep an estimate of the learner's current state of knowledge, and the dynamics of that state (e.g., forgetting). Keeping track of the learner's state allows for incrementally solving the optimization problem to take advantage of the words to which the learner had been exposed previously. Perhaps the simplest way in which this can be accomplished within the context of the proposed model is to directly associate the learner's state with the total number of past exposures to each word, and add the number of past exposures to all additional exposures in the objective of Equation 8.6. This would not, however, account for the dynamics of long-term re-

tention. Additionally, incorporating the learner's state in the optimization problem for generating mixed-language content would allow to generate progressively "denser" texts (i.e., larger fraction of switched words), as words that the learner had already mastered may appear in contexts that aid in acquiring new words, without detrimental effects.

- **Interest model** As shown in literature, attention and focus during reading is an important factor in incidental word acquisition [49, 156, 166]. In generating mixed-language content, a practical solution may take the form of a recommendation engine. In other words, instead of forcing the reader to read specific articles, the system may take into account the user's preferences for certain topics, and their knowledge state, to present a set of articles that will be instructive and interesting to the learner.
- **Assessment and feedback** Keeping the learner's state without active measurement (e.g., via assessments or quizzes) will likely lead to a divergence in its estimate over an extended period of time. As such, extensive reading must be intermittently mixed with assessment in order to update the model's estimate of the learner's state.

## CHAPTER 9

# LEARNING CONTENT REPRESENTATION FOR PERSONALIZED LESSON SEQUENCING

### 9.1 Introduction

As traditional learning resources like textbooks and tests become more readily off-loaded to the web, the rich student interaction data that comes with them forms the foundation for building data driven learning tools that adapt to the learners and to the changing landscape of the learning content. In this chapter we propose using course interaction data to jointly induce a continuous representation (embedding) of an entire course: its students, lessons and assessments. The proposed representation is able to naturally capture the dynamics of learning as movement through a vector field induced by the lesson modules. We demonstrate the effectiveness of this representation at the task of personalized curriculum recommendation and assessment outcome prediction, via a large scale empirical study on data collected from over a thousand classrooms during a period of five months by Knewton – an adaptive learning technology company.

As more of the students' interactions in online course platforms are logged, datasets in the form of *access traces* can be leveraged to understand the effectiveness of the individual learning modules. Access traces are personal timelines that log the sequence of learning modules and assessments that a student completed, together with the response outcomes (e.g., binary pass/fail). These access traces have the form *Student A completed Lesson B* and *Student C passed assessment D*. *Assessments* are content modules with pass-fail results that test student skills; for example, a true-or-false question halfway through a video lecture. Our task

in this chapter is to (i) induce a shared, dynamic representation (embedding) of the classroom — students, lesson modules and assessments — based on the observations of students’ access traces alone and (ii) utilize this representation for the purpose of predicting students’ outcomes and recommending lesson sequences. We evaluate our model on a set of progressively challenging synthetic scenarios, as well as large-scale real data from Knewton, an education technology company that offers personalized recommendations and activity analytics for online courses [91]. The data set consists of 2.18 million access traces from over 7,000 students, recorded in 1,939 classrooms over a combined period of 5 months.

## 9.2 Our Contributions

Our task can be categorized under an existing body of research known as *knowledge tracing*, which has a long history going back to the 1990s [44, 92]. The goal of knowledge tracing is to extract a signal of students’ mastery of his or her underlying skills through the observations of the assessment response outcomes (i.e., correctness of students’ answers). The key distinction of our model from traditional knowledge tracing [40], is in that our model does not rely on an expert’s annotation of the underlying skills, and discovers the underlying skills automatically from access traces. To the best of our knowledge, we are the first to identify and tackle several key challenges faced by observational studies in this space:

### 1. Offline Evaluation of Content Recommendation Policies:

Observational studies, such as the one we conduct in this chapter, are notoriously challenging for drawing causal inferences. Our goal in this chapter is

to evaluate the effectiveness of the proposed model in its ability to identify effective learning paths through the educational content. Observational data of students' paths, however, is heavily confounded by external factors besides the effectiveness of a taken path, such as students' own preferences towards certain learning content, or the directions given by the course instructor or an underlying recommendation engine (all three are factors in the data collected by Knewton). In this chapter, we develop a principled technique for assessing the performance of our model in recommending productive learning paths by de-biasing the observational data to account for a significant number of potential confounders.

2. **Model Sensitivity to Size of Training Data:** The performance of the induced embedding is a function of the number and the lengths of the access traces that were used in estimating the parameters of the embedding. We evaluate the impact of the training set size on the performance in predicting held-out assessment outcomes.

### 9.3 Embedding Model

We now describe the Latent Skill Embedding, a probabilistic model that places students, lessons, and assessments in a joint semantic space that we call the *latent skill space*. Students have trajectories through the latent skill space, while assessments and lessons are placed at fixed locations. Formally, a student is represented as a set of  $d$  latent skill levels  $\mathbf{s} \in \mathbb{R}_+^d$ ; a lesson module is represented as a vector of skill gains  $\ell \in \mathbb{R}_+^d$  and a set of prerequisite skill requirements  $\mathbf{q} \in \mathbb{R}_+^d$ ; an assessment module is represented as a set of skill requirements  $\mathbf{a} \in \mathbb{R}_+^d$ .

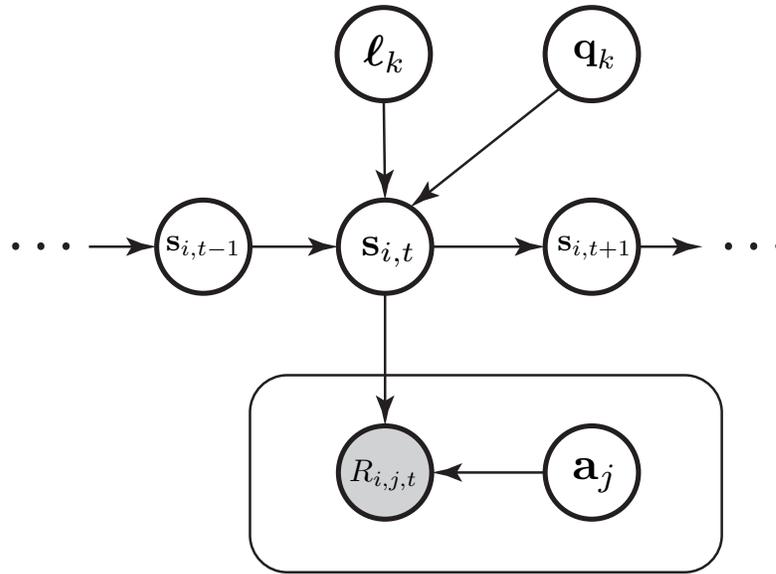


Figure 9.1: A graphical model of student learning and testing, i.e., a continuous state space Hidden Markov Model with inputs and outputs.  $\mathbf{s}$  = student knowledge state,  $\ell$  = lesson skill gains,  $\mathbf{q}$  = lesson prerequisites,  $\mathbf{a}$  = assessment requirements, and  $R$  = result.

Students interact with lessons and assessments in the following way. First, a student can be tested on an assessment module with a pass-fail result  $R \in \{0, 1\}$ , where the likelihood of passing is high when a student has skill levels that exceed the assessment requirements and vice-versa. Second, a student can work on lesson modules to improve skill levels over time. To fully realize the skill gains associated with completing a lesson module, a student must satisfy prerequisites (only partly fulfilling the prerequisites will result in relatively smaller gains, see Equation 9.4 for details). Time is discretized such that at every timestep  $t \in \mathbb{N}$ , a student completes a lesson and may complete zero or many assessments. The evolution of student knowledge can be formalized as the graphical model in Fig. 9.1, and the following subsections elaborate on the details of this model.

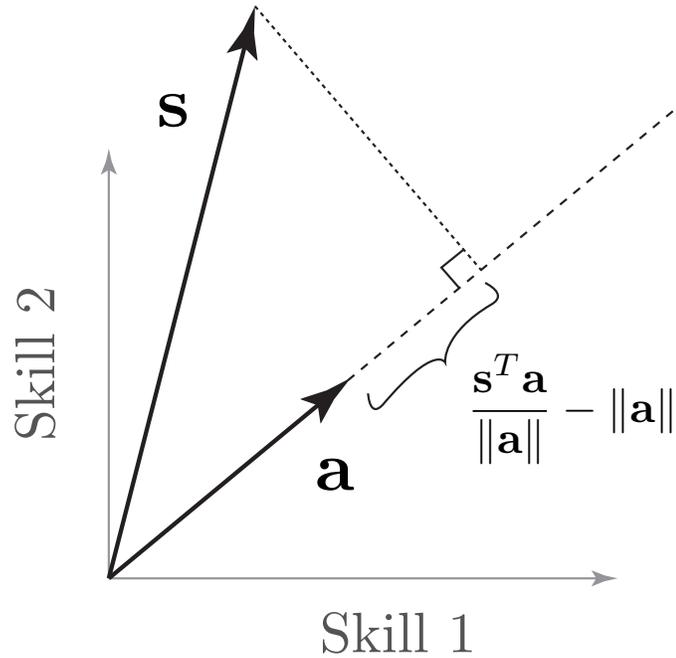


Figure 9.2: Geometric intuition underlying the parametrization of the assessment result likelihood (Equation 9.1). Only the length of the projection of the student’s skills  $\mathbf{s}$  onto the assessment vector  $\mathbf{a}$  affects the pass likelihood of that assessment, meaning only the “relevant” skills (with respect to the assessment) should determine the result.

### 9.3.1 Modeling Assessment Results

For student  $\mathbf{s}$ , assessment  $\mathbf{a}$ , and result  $R$ ,

$$R \sim \text{Bernoulli}(\phi(\Delta(\mathbf{s}, \mathbf{a}))) \tag{9.1}$$

where  $\phi$  is the logistic function and

$$\Delta(\mathbf{s}, \mathbf{a}) = \frac{\mathbf{s}^T \mathbf{a}}{\|\mathbf{a}\|} - \|\mathbf{a}\| + \gamma_s + \gamma_a \tag{9.2}$$

$\mathbf{s}$  and  $\mathbf{a}$  are constrained to be non-negative (for details see the Parameter Estimation section). A pass result is indicated by  $R = 1$ , and a fail by  $R = 0$ . The term  $\frac{\mathbf{s} \cdot \mathbf{a}}{\|\mathbf{a}\|}$  can be rewritten as  $\|\mathbf{s}\| \cos(\theta)$ , where  $\theta$  is the angle between  $\mathbf{s}$  and  $\mathbf{a}$ ; it can be interpreted as “relevant skill”. The term  $\|\mathbf{a}\|$  can be interpreted as general (i.e., not concept-specific) assessment difficulty. The expression  $\frac{\mathbf{s}^T \mathbf{a}}{\|\mathbf{a}\|} - \|\mathbf{a}\|$  is visualized in Fig. 9.2. The bias term  $\gamma_s$  is a student-specific term that captures a student’s general (assessment-invariant and time-invariant) ability to pass; it can be interpreted as a measure of how well the student guesses correct answers. The bias term  $\gamma_a$  is a module-specific term that captures an assessment’s general (student-invariant and time-invariant) difficulty.  $\gamma_a$  differs from the  $\|\mathbf{a}\|$  difficulty term in that it is not bounded; see the Parameter Estimation section for details. These bias terms are analogous to the bias terms used for modeling song popularity in [33]. Our choice of  $\Delta$  differs from traditional multi-dimensional item response theory, which uses  $\Delta(\mathbf{s}, \mathbf{a}) = \mathbf{s}^T \mathbf{a} + \gamma_a$  where  $\mathbf{s}$  and  $\mathbf{a}$  are not bounded (although in practice, suitable priors are imposed on these parameters).

### 9.3.2 Modeling Student Learning from Lessons

For student  $\mathbf{s}$  who worked on a lesson with skill gains  $\ell$  and no prerequisites at time  $t + 1$ , the updated student state is

$$\mathbf{s}_{t+1} \sim \mathcal{N}(\mathbf{s}_t + \ell, \Sigma) \quad (9.3)$$

where the covariance matrix  $\Sigma = I_d \sigma^2$  is diagonal. For a lesson with prerequisites  $\mathbf{q}$ ,

$$\mathbf{s}_{t+1} \sim \mathcal{N}(\mathbf{s}_t + \ell \phi(\Delta(\mathbf{s}_t, \mathbf{q})), \Sigma) \quad (9.4)$$

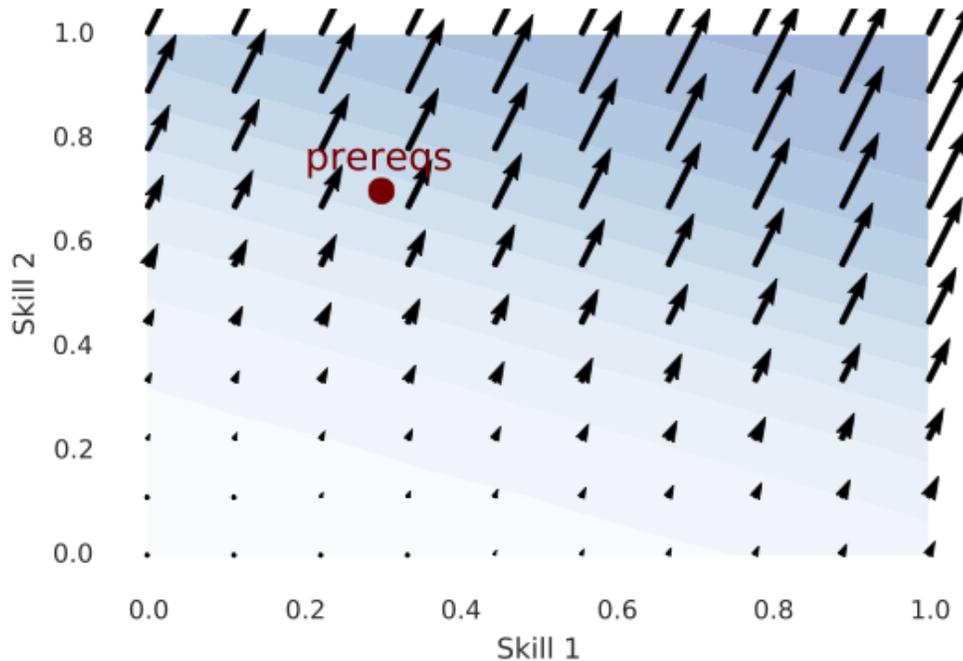


Figure 9.3: The vector field of skill gains for a lesson with skill gains  $\ell = (0.5, 1)$  and prerequisites  $\mathbf{q} = (0.7, 0.3)$ . Contours are drawn for varying update magnitudes. A student can compensate for lack of prerequisites in one skill through excess strength in another skill, but the extent to which this trade-off is possible depends on the relative weights of the prerequisites.

where  $\phi$  is the logistic function and  $\Delta(\mathbf{s}_t, \mathbf{q}) = \frac{\mathbf{s}_t^T \mathbf{q}}{\|\mathbf{q}\|} - \|\mathbf{q}\|$ . The intuition behind this equation is that the skill gain from a lesson should be weighted according to how well a student satisfies the lesson prerequisites. A student can compensate for lack of prerequisites in one skill through excess strength in another skill, but the extent to which this trade-off is possible depends on the relative weights of the prerequisites. The same principle applies to satisfying assessment skill requirements. With prerequisites, the vector field of skill gains is non-uniform (without prerequisites, it is uniform); for example, see Fig. 9.3.

Our model differs from [101] in that we explicitly model the effects of prerequisite knowledge on gains from lessons. Lan et al. model gains from a lesson as an affine transformation of the student’s knowledge state.

## 9.4 Parameter Estimation

We compute MAP estimates of model parameters  $\Theta$  by maximizing the following objective function:

$$\begin{aligned}
 L(\Theta) = & \sum_{\mathcal{A}} \log (\mathbb{P}[R \mid \mathbf{s}_t, \mathbf{a}, \gamma_s, \gamma_a]) \\
 & + \sum_{\mathcal{L}} \log (\mathbb{P}[\mathbf{s}_{t+1} \mid \mathbf{s}_t, \ell, \mathbf{q}]) - \beta \cdot \lambda(\Theta)
 \end{aligned} \tag{9.5}$$

where  $\mathcal{A}$  is the set of assessment interactions,  $\mathcal{L}$  is the set of lesson interactions,  $\lambda(\Theta)$  is a regularization term that penalizes the  $L_2$  norms of embedding parameters (not bias terms), and  $\beta$  is a regularization parameter. Non-negativity constraints on embedding parameters (not bias terms) are enforced.

$L_2$  regularization is used to penalize the size of embedding parameters to prevent overfitting. The bias terms are not bounded or regularized. This allows  $-\|\mathbf{a}\| + \gamma_a$  to be positive for assessment modules that are especially easy, and  $\frac{\mathbf{s} \cdot \mathbf{a}}{\|\mathbf{a}\|} + \gamma_s$  to be negative for students who fail especially often. We solve the optimization problem with box constraints using the L-BFGS-B [212] algorithm. We randomly initialize parameters and run the iterative optimization until the relative difference between consecutive objective function evaluations is less than  $10^{-3}$ . Averaging validation accuracy over multiple runs during cross-validation reduces sensitivity to the random initializations (since the objective function is

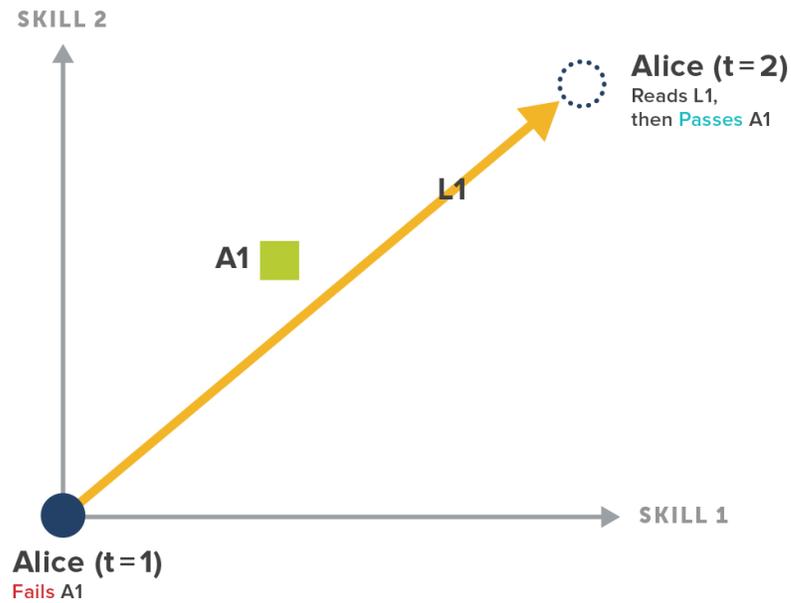


Figure 9.4: An extremely simple embedding

non-convex).

## 9.5 Experiments on Synthetic Data

To verify the correctness of our model and to illustrate the properties of the embedding geometry that the model captures, we conducted a series of experiments on small, synthetically-generated interaction histories. Each scenario is intended to demonstrate a different feature of the model (e.g., recovering student knowledge and assessment requirements in the absence of lessons, or recovering sensible skill gain vectors for different lessons). For the sake of simplicity, the embeddings do not use bias terms. The scenarios shown next are annotated versions of plots made by our embedding software.

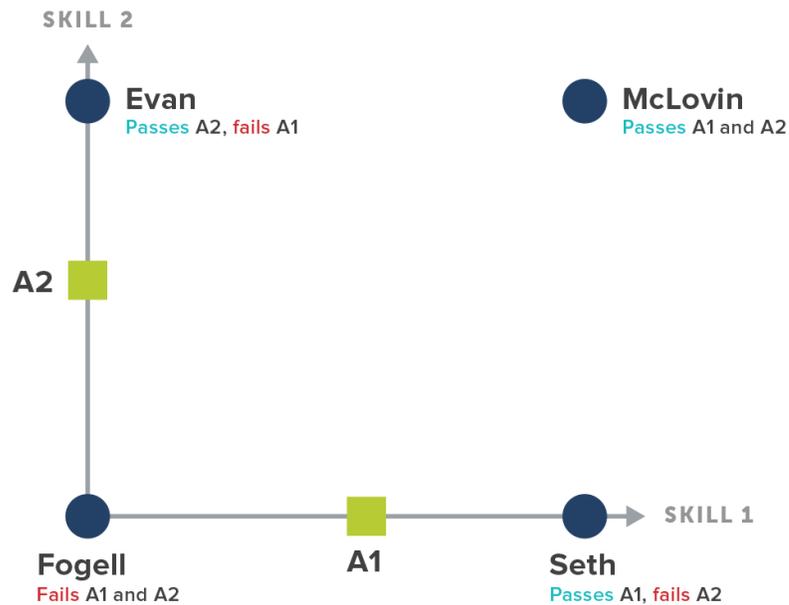


Figure 9.5: A two-dimensional embedding without lessons

Fig. 9.4 demonstrates an extremely simple embedding. The key observation here is that the model recovered positive skill gains for lesson L1, and “correctly” arranged Alice and assessment A1 in the latent space. Initially, Alice fails A1, so her skill level is behind the requirements of A1. After completing L1, Alice passes A1, indicating that her skill level has probably improved past the requirements of A1. Note that this scenario could have been explained with only one latent skill.

Fig. 9.5 depicts a two-dimensional embedding, where an intransitivity in assessment results requires more than one latent skill to explain. The key observation here is that the assessments are embedded on two different axes, meaning they require two completely independent skills. This makes sense, since student results on A1 are uncorrelated with results on A2. Fogell fails both assessments, so his skill levels are behind the requirements for A1 and A2. McLovin passes

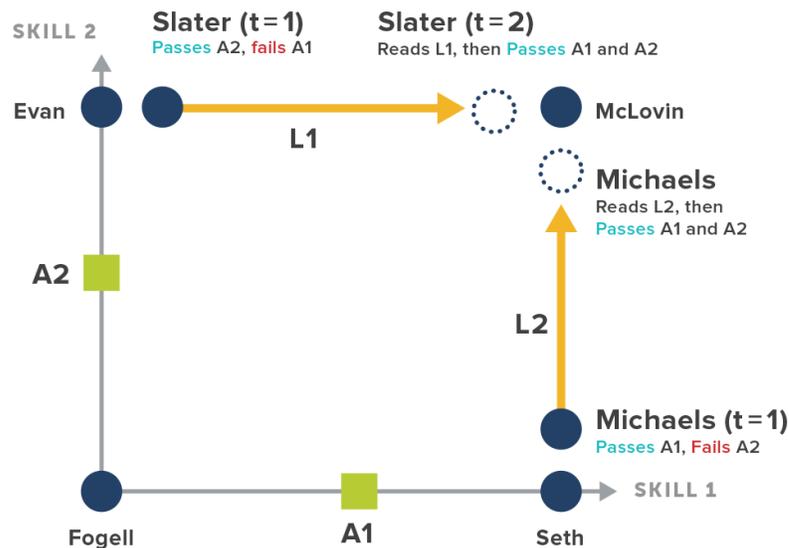


Figure 9.6: A two-dimensional embedding with lessons, without prerequisites

both assessments, so his skill levels are beyond the requirements for A1 and A2. Evan and Seth are each able to pass one assessment but not the other. Since the assessments have independent requirements, this implies that Evan and Seth have independent skill sets (i.e., Evan has enough of skill 2 to pass A2 but not enough of skill 1 to pass A1, and Seth has enough of skill 1 to pass A1 but not enough of skill 2 to pass A2).

In Fig. 9.6, we replicate the setting in Fig. 9.5, then add two new students Slater and Michaels, and two new lesson modules L1 and L2. Slater is initially identical to Evan, while Michaels is initially identical to Seth. Slater reads lesson L1, then passes assessments A1 and A2. Michaels reads lesson L2, then passes assessments A1 and A2. The key observation here is that the skill gain vectors recovered for the two lesson modules are orthogonal, meaning they help students

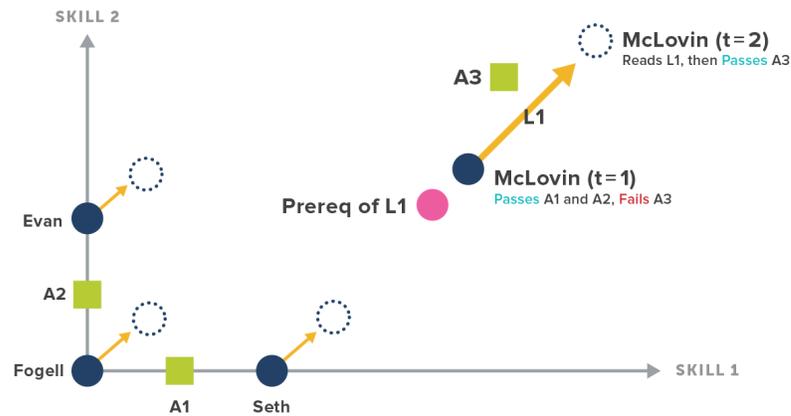


Figure 9.7: A two-dimensional embedding with lessons and prerequisites

satisfy completely independent skill requirements. This makes sense, since initially Slater was lacking in Skill 1 while Michaels was lacking in Skill 2, but after completing their lessons they passed their assessments, showing that they gained from their respective lessons what they were lacking initially.

In Fig. 9.7, we replicate the setting in Fig. 9.5, then add a new assessment module A3 and a new lesson module L1. All students initially fail assessment A3, then read lesson L1, after which McLovin passes A3 while everyone else still fails A3. The key observation here is that McLovin is the only student who initially satisfies the prerequisites for L1, so he is the only student who realizes significant gains from taking L1.

## 9.6 Experiments on Online Course Data

We use data processed by Knewton, an adaptive learning technology company. Knewton's infrastructure uses student-content access traces to generate personalized recommendations and activity analytics for partner organizations with online learning products. The data describes interactions between college students and two science textbooks. The Book A data set was collected from 869 classrooms from January 1, 2014 through June 1, 2014. It contains 834,811 interactions, 3,471 students, 3,374 lessons, 3,480 assessments, and an average assessment pass rate of 0.712. The paths that students take are biased by direction from instructors, a recommender system, and the sequence of chapters in the textbook. The Book B data set was collected from 1,070 classrooms from January 1, 2014 through June 1, 2014. It contains 1,349,541 interactions, 3,563 students, 3,843 lessons, 3,807 assessments, and an average assessment pass rate of 0.693.

Both data sets are filtered to eliminate students with fewer than five lesson interactions and content modules with fewer than five student interactions. To avoid spam interactions and focus on the outcomes of initial student attempts, we only consider the first interaction between a student and an assessment (subsequent interactions between student and assessment are ignored).

### 9.6.1 Assessment Result Prediction

We evaluate the embedding model on the task of predicting results of held-out assessment interactions, and compare it to three benchmark models: the one- and two-parameter logistic item response theory models, and a two-dimensional

	Model			Book A		Book B	
	$\vec{\ell}$	$\vec{q}$	$\gamma$	Test	Validation	Test	Validation
<b>1</b>	N	N	N	0.673	0.614 ± 0.015	0.614	0.644 ± 0.015
<b>2</b>	N	N	Y	0.818	0.753 ± 0.020	0.788	0.821 ± 0.021
<b>3</b>	Y	N	N	0.692	0.624 ± 0.019	0.630	0.662 ± 0.023
<b>4</b>	Y	N	Y	0.798	0.761 ± 0.016	0.775	0.808 ± 0.020
<b>5</b>	Y	Y	N	0.724	0.625 ± 0.021	0.629	0.643 ± 0.018
<b>6</b>	Y	Y	Y	0.811	0.756 ± 0.018	0.785	0.823 ± 0.021
<b>7</b>	1PL IRT			0.812	0.761 ± 0.016	0.778	0.812 ± 0.019
<b>8</b>	2PL IRT			0.780	0.708 ± 0.011	0.686	0.690 ± 0.022
<b>9</b>	2D MIRT			0.817	0.732 ± 0.012	0.776	0.796 ± 0.018

Table 9.1: Test AUC, validation AUC, and standard error of validation AUC for variants of the LSE model and benchmark IRT models.

item response theory model. The 1PL IRT model, also known as the Rasch model, has the following assessment pass likelihood:  $\mathbb{P}[R = 1] = \phi(\theta_i - \beta_j)$  for student  $i$  and item  $j$ , where  $\theta$  is student proficiency and  $\beta$  is item difficulty, and  $\phi$  is the logistic link function [150]. The 2PL model extends the likelihood as follows:  $\mathbb{P}[R = 1] = \phi(\alpha_j(\theta_i - \beta_j))$ , where  $\alpha$  is the item discriminability [110]. The 2D MIRT model, which is a multi-dimensional generalization of 2PL, has the following pass likelihood:  $\mathbb{P}[R = 1] = \phi(\mathbf{u}_i^T \mathbf{v}_j + \mu_j)$ , where  $\mathbf{u}$  are the student factors,  $\mathbf{v}$  are the item factors, and  $\mu$  is the item offset [154]. Note that we have not explicitly included Bayesian Knowledge Tracing as a benchmark model since it requires content modules to be annotated with concept tags, while the Latent Skill Embedding does not.

We use ten-fold cross-validation to select the regularization parameter  $\beta$  and learning update variance  $\sigma^2$  for the embedding model (see Fig. 9.8 for the exploration on Book A), as well as regularization parameters for the benchmark IRT models. On each fold, we train on the full histories of 90% of students and the truncated histories of 10% of students, and validate on the assessment

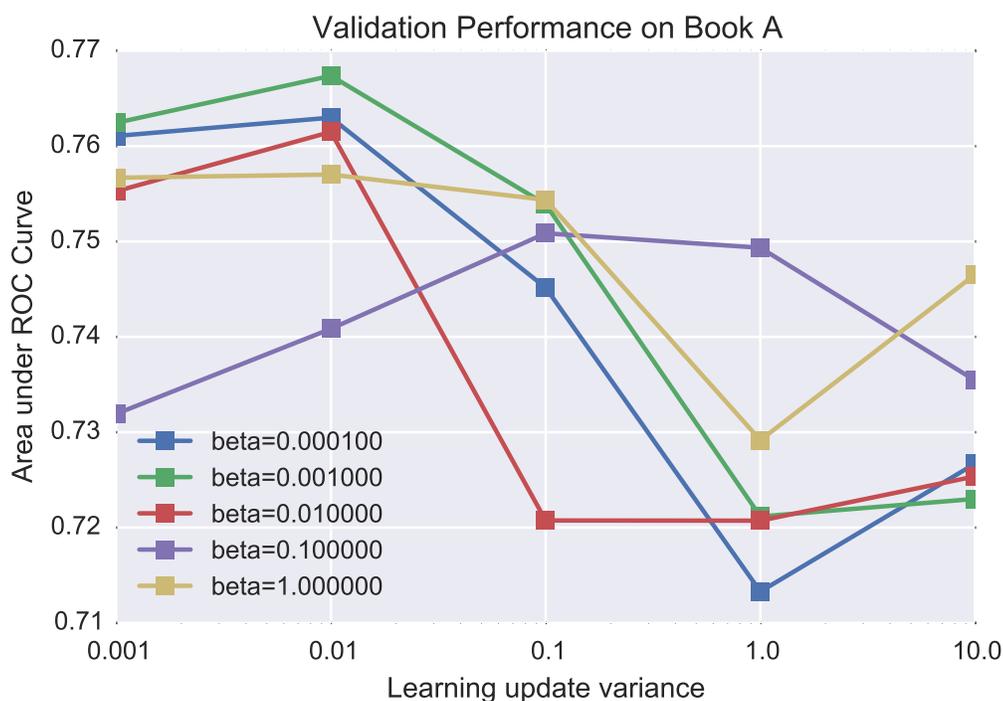


Figure 9.8: We explore the parameter space of the two-dimensional embedding with prerequisites and bias terms by doing a grid search on  $(\sigma^2, \beta)$ .

interactions immediately following the truncated histories. Truncations are made just before the last assessment interactions for each student (maximizing the size of the training set). We have also examined the effect of randomizing the truncation in student histories, and find no substantial changes to our results.

After selecting model hyperparameters using cross-validation, we evaluate the models on a held-out test set of students (20% of the students in the complete data set) that was not visible during the earlier parameter selection phase. The same truncation method is used for evaluation on the test set. Our performance metric is area under the ROC curve (AUC), which measures the discriminative ability of a binary classifier that assigns probabilities to class membership.

**Lesion Analysis** To gain insight into which components of the embedding model contribute most to its predictive power, we conduct a lesion analysis. For the sake of simplicity, we restrict ourselves to using a two-dimensional embedding (later, we describe the effect of varying the embedding dimension  $d$ ). We start with an embedding model that ignores lesson interactions and does not use bias terms. We then gradually add components to the embedding model to examine their effects on prediction AUC. Specifically, we evaluate embeddings with and without lesson parameters  $\ell$ , prerequisite parameters  $\mathbf{q}$  for lessons, and bias terms  $\gamma$ . Each variant of the model corresponds to a row in Table 9.6.1.

From these results, we observe the following: including bias terms in the assessment result likelihood (Equation 9.1) gives a large and statistically significant performance gain ( $p < 0.0003$  for the standard t-test comparing validation AUCs of row 5 vs. 6 on Book A); an embedding with lesson prerequisites and bias terms performs comparably to the best benchmark IRT model.

**Effect of Embedding Dimension** In other experiments, we explored the parameter space of the embedding model by varying the regularization constant  $\beta$  and embedding dimension  $d$ . Not explicitly shown are the results for changing  $d$ . In summary, we find that increasing embedding dimension  $d$  substantially improved performance for embedding models without bias terms, but that it has little effect on performance for embeddings with bias terms. The former is expected, since the embedding itself must be used to model general student passing ability and general assessment difficulty.

**Sensitivity Analysis** We perform a sensitivity analyses on Book A and observe the following: prediction AUC is most affected by a student's recent history (see

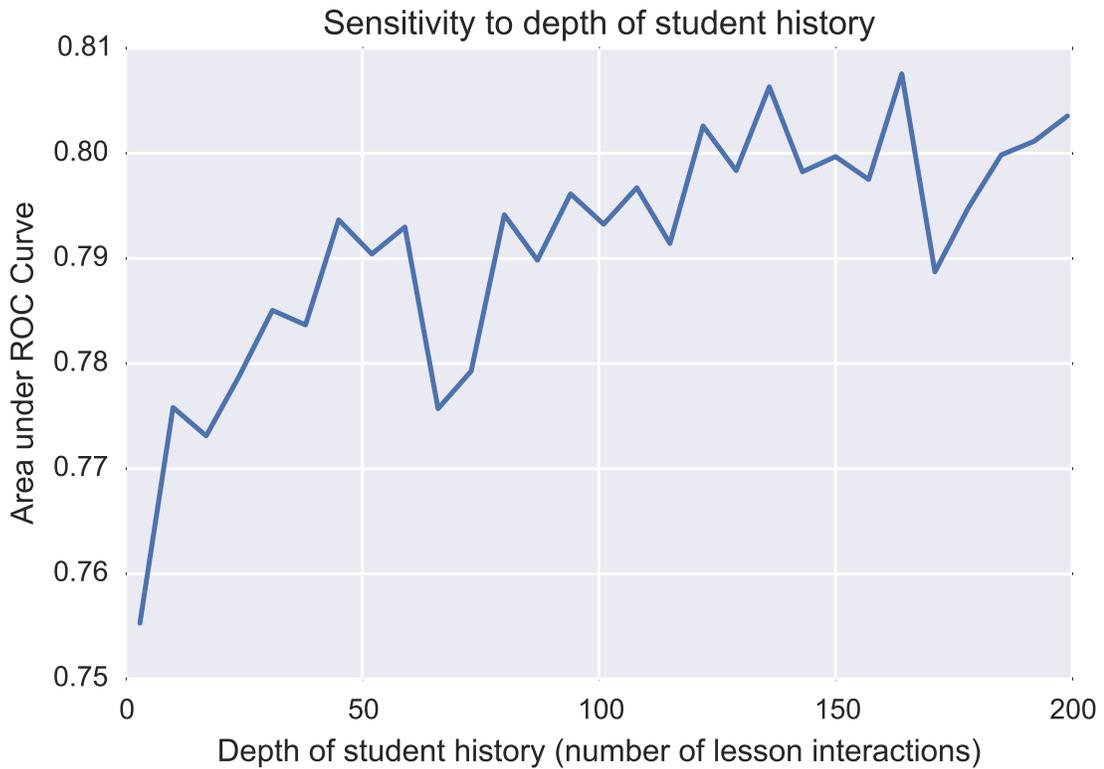


Figure 9.9: Sensitivity of validation AUC to the “depth” of a student’s history (from  $t = T - depth$  to  $t = T$ ). A student’s recent history is most helpful for predicting assessment results, which we observe in the plateauing of the curve as we gradually include interactions from the students far past.

Fig. 9.9). This findings lead to a qualitative insight regarding model performance: for a course offered regularly (e.g., over several semesters), the model will improve steadily as log data is collected from students who complete the course.

## 9.6.2 Lesson Sequence Discrimination

The ability to predict future performance of students on assessments, while a useful metric for evaluating the learned embedding, does not address the more

important task of adaptive tutoring via customized lesson sequence recommendation. We introduce a surrogate task for evaluating the sequence recommendation performance of the model based entirely on the observational data of student interactions, by assessing the model’s ability to recommend “productive” paths amongst several alternatives.

**Bubbles as Experimental Evidence** The size of the data set creates a unique opportunity to leverage the variability in learning paths to simulate the setting of a controlled experiment. For this evaluation, we use a larger version of the Book A data set, containing 14,707 students and 14,327 content modules. We find that the data contains many instances of student paths that share the same lesson module at the beginning and the same assessment module at the end, but contain different lessons along the way. We call these instances *bubbles*, for example see Fig. 9.10, which present themselves as a sort of experimental evidence on the relative merits of two different learning progressions. We can thus use these *bubbles* to evaluate the ability of an embedding to recommend a learning sequence that leads to success, as measured by the relative performance of students who take the recommended vs. the not-recommended path to the assessment module at the end of the *bubble*.

We use the full histories of 70% of students to embed lesson and assessment modules, then train on the histories of held-out students up to the beginning of a *bubble*. The lesson sequence for a student is then simulated over the initial student embedding, using the learning update (Equation 9.4) to compute an expected student embedding at the end of the *bubble* (which can be used to predict the passing likelihood for the final assessment using Equation 9.1). The path that leads the student to a higher pass likelihood on the final assessment

is the “recommended” path. Our performance measure is  $\mathbb{E} \left[ \frac{\mathbb{E}[R'] - \mathbb{E}[R]}{\mathbb{E}[R]} \right]$ , where  $R' \in \{0, 1\}$  is the outcome at the end of the recommended path and  $R \in \{0, 1\}$  is the outcome at the end of the other path (0 is failing and 1 is passing). This measure can be interpreted as “expected gain” (averaged over many *bubbles*) from taking recommended paths, or how “successful” the paths recommended by the model are when compared to the alternative.

**Propensity Score Matching** This observational study is potentially confounded by many hidden variables. For example, it may be that one group of students systematically takes recommended paths while another group of students does not, leading to results at the end of a *bubble* that are mostly dictated by the teachers directing the groups, or other student-specific hidden factors, rather than path quality. To best approximate the settings of a randomized controlled trial in our observational study, we use the standard *propensity score matching* approach for de-biasing observational data [160, 31]. The key idea behind *propensity score matching* is to subset the observed data in a way that balances the distribution of the features (“hidden variables”) describing subjects in the two conditions, as it would be expected in a randomized experiment. The validity of any conclusion drawn from the observational data de-biased in this way hinges on the assumption that all confounding variables that determine self-selection have been accounted for in the features prior to matching. In this study, we hypothesize that the set of all lesson modules and assessment modules (with outcomes) that the learner attempted throughout his or her duration in the online system is sufficient to compensate for any self-selection in the taken learning paths. Formally, we represent learners in a feature space  $X$  such that  $X_{ij} \in \{-1, 0, 1\}$ , where  $X_{ij} = 1$  if student  $i$  passed module  $j$  (lessons are always “passed”),  $X_{ij} = 0$

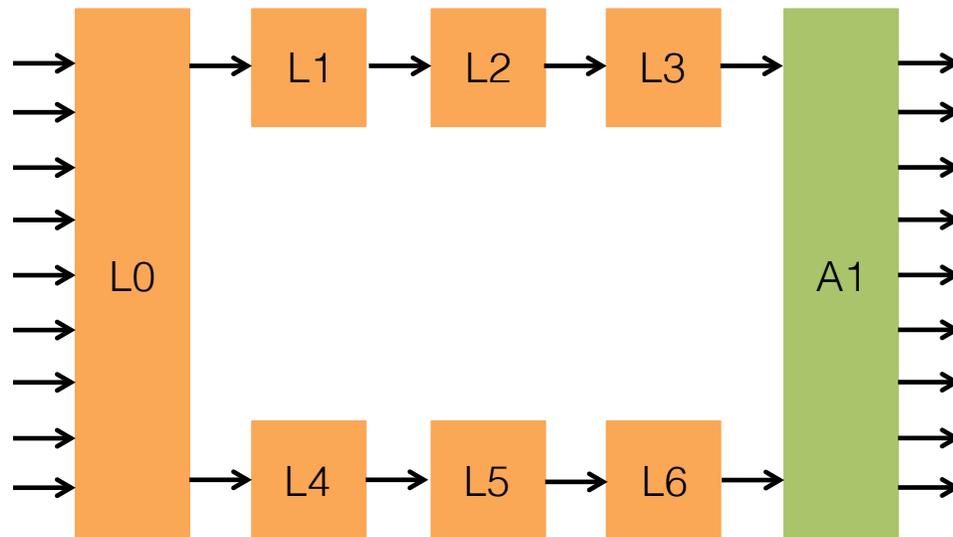


Figure 9.10: A schematic diagram of a *bubble*, where a group of students converges on a lesson, splits off into two different lesson sequences, then converges on the same assessment.

if student  $i$  has not completed module  $j$ , and  $X_{ij} = -1$  if student  $i$  failed module  $j$ .

We use PCA to map  $X$  to a low-dimensional feature space where students are described by 1,000 features, which capture 80% of the variance in the original 14,327 features. A logistic regression model with  $L_2$  regularization is used to estimate the probability of a student following the recommended branch of a *bubble*, i.e., the *propensity score*, given the student features (the regularization constant is selected using cross-validation to maximize average log-likelihood on held-out students). Within each *bubble*, students who took their recommended branch are matched with their nearest neighbors (by absolute difference in *propensity scores*) from the group of students who did not take their recommended branch. Matching is done with replacement (so the same student can be selected as a nearest neighbor multiple times) to improve matching quality, trading off bias for variance. Multiple nearest neighbors can be matched (we examine the

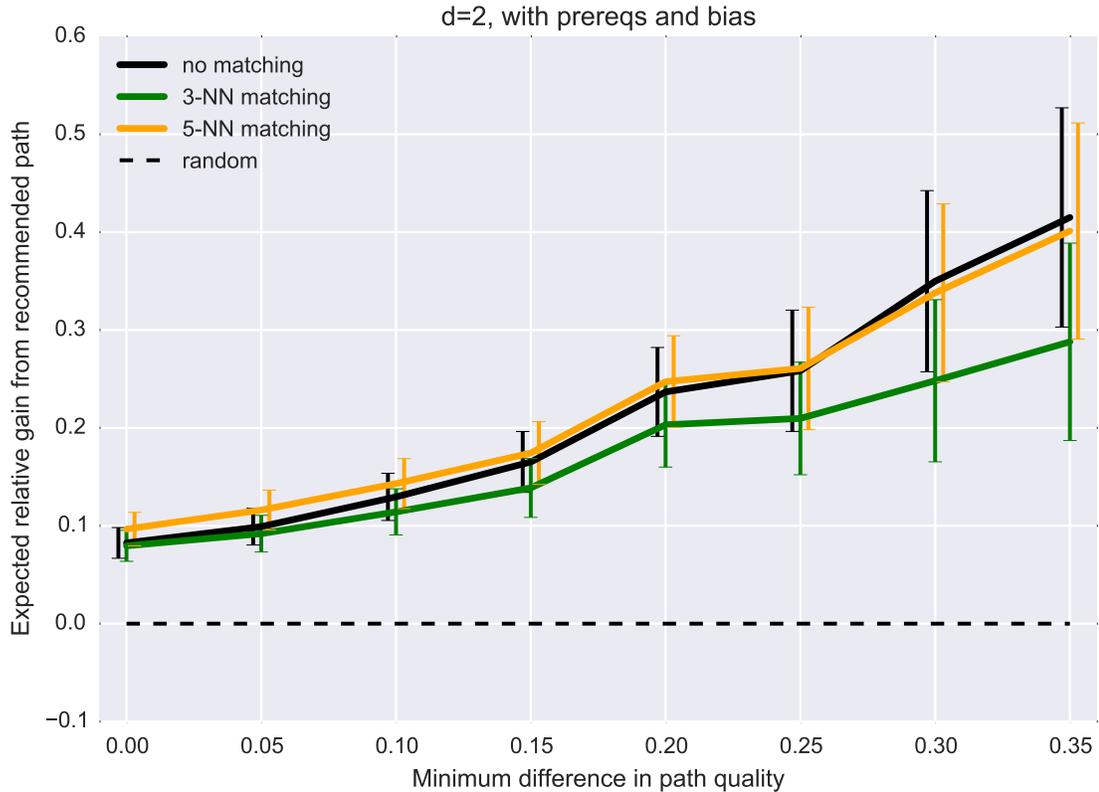


Figure 9.11: The  $x$ -axis represents a threshold on absolute difference between pass rates of the two *bubble* paths. *Bubbles* are filtered to meet the following criteria: at least ten students take each branch, each branch must contain at least two lessons, and both branches must contain the same number of lessons. The error bars represent standard error, and their  $x$ -coordinates are slightly perturbed so the error bars for different curves can be distinguished.

effect of varying  $k$ ), trading off variance for bias.

**Results** Fig. 9.11 shows the results of the experiment, showing by how much students gain by following the path recommended by our embedding. We use the same embedding configuration as in row 6 of Table 9.6.1, which uses prerequisites

and bias terms in a two-dimensional embedding model with lesson. Naturally, our evaluation metric of gain in the pass rate from following a recommended path would depend strongly on the relative merits of the recommended and alternative paths. We therefore plot the gain that the recommended path achieves in relation to the difference in path quality, as measured by the absolute difference in pass rates between the two paths. Fig. 9.11 shows that the model generally able to recommend more successful paths, and this finding is robust to the choice of nearest neighbors  $k$  used during propensity matching. As expected, the effect of the system recommendation is larger when there is a significant difference between the quality of the two paths.

## 9.7 Conclusion

In this chapter, we addressed the problem of learning a joint representation (embedding) of students and content from students' interactions with the content (i.e., access traces). Using synthetic and real-world datasets, we have shown that this representation is capable of capturing the dynamics of students' learning. Our second contribution is in developing an offline methodology for evaluating the effectiveness of this model in predicting effective learning paths to the students. The enabling factor for developing this offline methodology is the large number of logged interactions that allows us to discover nearly matching groups for approximating the setting of a randomized experiment. However, while the dataset used in this chapter is vast in the number of students and interactions, the content across all interactions is limited to only two textbooks. As textbooks typically explain each concept once (i.e., in one place), the consequence is a limit on the diversity in the paths that students take within that

content, in effect limiting the model's capacity to estimate the utility and quality of the different learning modules. In the next chapter, we explore the potential of scaling learning content by leveraging the vastness and the diversity of the web. The distinguishing characteristic of the web is its diversity of explanations and assumptions, i.e., the same concepts may be explained in different ways and assume different prerequisites, potentially resonating with learners of different backgrounds and needs.

## CHAPTER 10

### CURATING TARGETED LEARNING PATHS THROUGH THE WEB

#### 10.1 Introduction

In this chapter, we look towards the web as a vast source of diverse learning material. While physical textbooks and classrooms traditionally assumed the role of knowledge curators, they also present a bottleneck in today's rapidly growing web of up-to-date technical and academic content — peer-reviewed articles, lecture notes, tutorials, slides etc — from academics and “citizen scientists” alike. An automatic approach for “weaving” natural curricular progressions through the web of such heterogeneous academic/educational content, we believe, will catalyze early and lifelong learning by creating more efficient and goal-oriented curricula targeted to the level of the audience.

The web is the only collection of resources today where attempting this task becomes meaningful and promising. The reason for this is that the web contains an extensive amount of diversity in its content, i.e., content that explains the same concepts but in many different ways. Naturally this diversity reflects the diversity of the people who create this content, their backgrounds, styles of learning and ways of thinking about complex concepts, which would naturally match learners with similar characteristics. We believe that this diversity can be leveraged to create learning pathways that are not bound to the traditional curricula that are often constrained for no better than a historical reason. We propose instead to optimize a curriculum directly for *what you want to know* given *what you already know*.

---

<sup>0</sup>This chapter has been adapted from the paper [98]

We propose to tackle the problem of *curriculum mining* on the web, which broadly, involves linking technical resources on the web to other resources that explain a subset of concepts that are assumed in the original document. We propose to decompose the task into (i) understanding what is *explained* and *assumed* in a document on the part of the the reader and (ii) use this document-level representation to sequence documents that guide the learner from their current state of knowledge towards their goal, for example, understanding a specific research paper or a set of lecture notes.

We propose a *term-centric* approach for inducing curricular relations between any pair of documents. Naturally, understanding a technical concept is more than being familiar with its surface term, and in this view an approach that operates at the level of individual terms may appear to be naïve. After all, to explain a new concept is to put together existing concepts in a novel way [201], and in the process introduce convenient nomenclature. However, we hypothesize, that by the virtue of seeking the shortest sequence of documents that “cover” (explain) multiple terms at once, the resulting bottleneck will implicitly “prefer” to link to prerequisite documents that introduce and explain whole concepts, i.e., groups of terms, as opposed to introducing terms one document at a time (an extreme example would be presenting a sequence of pages from a dictionary, each document defining a term independence; this is clearly undesirable). It will be our running assumption, that there exists a correlation between the knowledge of the terms and the understanding of the overarching concept.

Thus, to a first-order approximation, we model technical documents as “bags of terms”, and in the interest of tractability set forth the following set of modeling assumptions:

- **Assumption 1** A document is a bag-of-technical-terms (multiset) that is further partitioned into two multisets: *E* (*Explained*), *A* (*Assumed*) — corresponding to the role (aspect) of the term within the document:

**Explained:** The terms appear in the context that furthers the understanding of the concept corresponding to the term.

**Assumed:** The concept corresponding to the term is assumed to be familiar, and is required for understanding the context in which it appears.

- **Assumption 2** The degree of reliance on the knowledge of a particular term in the document is proportional to the frequency of the term in the *Assume* multiset, i.e., which concepts are fundamental to the understanding of the document, and which are auxiliary is reflected in the number of occurrences of the corresponding terms.

As an illustration, consider the following excerpt from Christopher Bishop's classic textbook *Machine Learning and Pattern Recognition* from the chapter that introduces the concept of *Expectation Maximization*:

#### **Expectation Maximization**

An elegant and powerful method for finding maximum likelihood solutions for models with latent variables is called the expectation maximization algorithm, or EM algorithm.

In the excerpt above, we solid-underline the terms that appear in the *Explained* aspect and dash-underline terms that appear in the *Assumed* aspect. Understanding the concept of *Maximum likelihood* is a prerequisite for understanding *Expectation Maximization*. It is no surprise that most resources that introduce the concept of *Expectation Maximization* implicitly assume that the reader is familiar

with *Maximum Likelihood*. Academic and educational literature is fraught with such implicit assumptions that may be challenging to unravel for a learner especially new to the area. Note that on the surface it may seem that detecting instances of explained terms in the text is an equivalent task to finding instances of term definitions – a well studied task – but it is not so. Especially in technical disciplines, explaining a concept requires much more than giving a definition. A document defining a term, may or may not actually explain the concept behind it. For example, a document may define a term to refresh the reader’s memory but otherwise assume the reader’s familiarity with it. On the other hand, a document may explain a term without ever giving a one-sentence definition.

Finally, the proposed dichotomy may appear as a gross oversimplification, ignoring the entire continuum of pragmatics between the two extremes. We argue that while binary term-level classification alone may not capture the fine-grained aspect of any one term, combining it with the context of the entire document, will enable us to unravel the prerequisite relationships between documents.

## 10.2 Related work

**Evidence of information overload in traditional textbooks** Formal study of textbook organization conducted by [3] on a corpus of textbooks from India quantitatively addresses the issue known as the “mentioning problem” [191], where “concepts are encountered before they have been adequately explained and forces students to randomly ‘knock around’ the textbook”. The work of [3] suggests that many traditional textbooks suffer from the resulting phenomenon of “information burden” and provide diagnostic metrics for evaluating it. A user

study conducted by [4], though limited to electronic textbooks, demonstrated the utility of a navigational aid that links concepts and terms within a textbook and allows the user to navigate according to own preferences. This suggests the potential utility of tools that expand such “navigational ability” outside textbooks.

**Attempts at manual curriculum curation** There have been at least two efforts that we are aware of, that attempts to manually create “paths” between a selected set of resources on the web — two educational start-ups, Metacademy [66], and Knewton [52]. While motivated by the same goal, we believe that manual web-scale curriculum curation is akin to the manually-curated directory of the web (not too different from the original Yahoo directory from the 1990s), i.e., offering poor scaling capability in the dynamic, growing landscape of educational content on the web.

**Attempts at automatic curriculum curation** Most relevant to our task is the work of [184] that attempt to infer prerequisite relationships between a pair of Wikipedia articles. They frame the problem of prerequisite prediction as “link-prediction” between a pair of pages using primarily graph-derived (e.g., hyperlink structure) and some content-derived features (e.g., article titles). In contrast to their approach, we do not assume any existing structure connecting the web resources (e.g., within Wikipedia), as the majority of the educational content on the web is unstructured. Our approach also naturally facilitates a scalable assimilation of new content, as we require only a document-scoped term-level classification, without needing to explicitly construct or update a prerequisite graph. Furthermore, we develop an approach for optimizing curricular paths using the proposed representation. More recent work of [107] develop a method

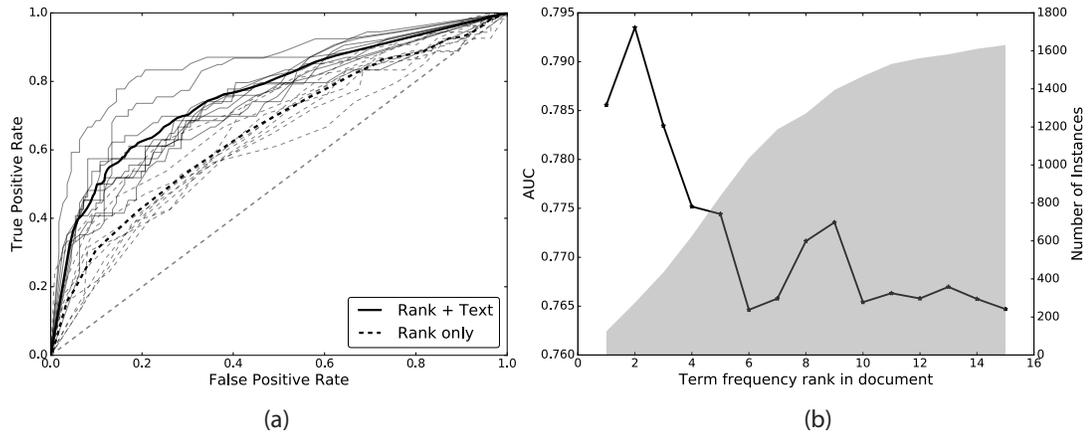


Figure 10.1: (a) ROC curves for the task of binary aspect classification. (b) AUC (left  $y$ -axis) of aspect classification for terms with a maximum document rank given on  $x$ -axis. Shaded region shows the number of terms up to the given maximum rank (right  $y$ -axis).

that does not rely on a manual annotation of the prerequisite relations as in [184], and instead uses the statistics of concept reference in a pair of pages to determine the prerequisite relation between them. Similar to [184], their focus is on the pairwise link prediction, in contrast to our goal of globally optimizing a learning curriculum.

### 10.3 Model

We model the problem of identifying the explained and assumed terms in a document as a term-level binary classification task, i.e each term in the document is classified into one of the two categories. Although simple from an implementation perspective, this task is made difficult by the lack of annotated data in this domain. In this chapter, we rely on (i) manual annotation of the term aspects

performed by us for one of the textbooks (Rice University’s statistics text) and (ii) explicit annotations from the index of Bishop’s Pattern Recognition and Machine Learning textbook that were made by the author of the text (the annotation is in the form of a location in the text where a particular concept is explained).

The Rice University’s *Online Statistics Education: An Interactive Multimedia Course of Study* textbook, from hereon referred to as STATSBOOK consists of a total of 112 units, with a median of 12.5 unique technical terms per unit, for a total of 339 different technical terms in the book. We scrape the text content of the book from the web, replace all mathematical formulae and symbols with special tokens, and manually annotate each technical term mention with its representative form from the index, i.e., *normally distributed* with *normal distribution*. Manual term annotation obviates the need for introducing a word-sense disambiguation component and additional errors. We process the PRML dataset in an identical manner.

Each technical term in every unit of the book was annotated with the binary  $\{explain, assume\}$  aspect, following the definitions outlined on the previous page. While for most terms, the application of these definitions is fairly unambiguous, for a significant number of term mentions, the aspects are not mutually exclusive, i.e., the term may be construed to belong to both aspects simultaneously. Often, in using (assuming) a term to explain a related concept, something about the assumed term is also explained as a side effect. The degree to which the explanation is distributed between the terms is difficult to judge objectively, and may vary between distinct mentions of the terms in different parts of the same document. We adopt a simple strategy for “breaking ties” in such cases: if we judge a term as having been *intended* to be explained in the given context by the

author, we mark it with the *explain* aspect, otherwise, the term is assumed to be *assumed*. In total across the entire STATSBOOK corpus, 1878 terms were annotated for their aspect (note that the same term appears in multiple documents with potentially different aspects), with a class ratio of 537 terms belonging to the *explain* and 1341 terms belonging to the *assume* aspect.

The PRML dataset contains a total of 3883 annotated terms, with 222 terms belonging to the *explain* and 3661 terms belonging to the *assume* aspect. The aspect of the term was determined from the index of the book, which explicitly specifies the pages where a term is explained.

A logistic regression model (LIBLINEAR [50] with default regularization parameter) was trained to predict a binary aspect of the terms and evaluated with 10-fold stratified cross-validation. A set of lexical and dependency features describing the context of each term (within a 1 sentence window), positional features describing the location of the term's mention within the document and sentences in which the term appeared, and the frequency rank of the term within the document were employed. We compare the performance of a classifier that uses all of these features with the one that uses only the rank. A classifier that is given rank as the only feature, will essentially learn a rank "threshold" that will decide the aspect of the term within the document, i.e., predict all terms above a certain rank as *explained*.

## Results

Figure 10.1(a) summarizes the performance of aspect prediction with the classifier trained using both linguistic and rank features (Rank+Text, AUC=0.76) versus a

classifier trained using only the rank (Rank only, AUC=0.66) for the STATSBOOK corpus. As expected, rank is predictive of the aspect, but contextual linguistic cues provide a significant boost.

Keeping our end goal in mind, under Assumption 2 stated in the introduction, we hypothesize that the frequency rank of the term in a document correlates with the degree to which a term is either assumed or explained in that document. In the downstream task of linking documents to their prerequisites, getting the aspects of the more frequent terms correct is arguably more important than of the terms that only appear once or twice. We evaluate the performance of our aspect classifier as a function of the term's rank. Figure 10.1(b) illustrates predictive performance (AUC) on a subset of the data stratified by the term's frequency rank. We observe a favorable trend in increased predictive performance for higher ranked terms. An obvious explanation is that more frequent terms accumulate a larger set of features describing them (since each mention of the term contributes its context features), effectively decreasing variance in the predictions.

### 10.3.1 Semi-supervised learning of term aspect

While textbooks in many disciplines are widely available and easily accessible, providing a valuable resource for training a model for predicting term aspect (i.e., *explanation* vs. *assumption*) in many domains, few of these textbooks provide labels of these aspects like the PRML textbook. Our key insight is that a textbook by its design encodes a very strong prior over the distribution of *assume* and *explain* labels for a single term, across the length of the book, which can be exploited to learn the labels simultaneously with the model. Specifically, concepts

tend to be explained primarily in one place (e.g., chapter) and assumed thereafter. This distributional assumption can be viewed as a constraint during training of the model. Specifically, consider a single term  $t_{ij}$  (word  $i$  in document  $j$ ). We denote  $\mathbf{x}_{ij}$  as a context feature of term  $t_{ij}$  in the document. In our implementation the features consist of co-occurring unigram, bigrams, grammatical role and various statistics, like frequency and rank of the term. The linguistic features are collected from within a window of 1 sentence of each mention of the term, and then aggregated across all mentions of that term in the document. We employ a logistic regression classifier to predict whether the term  $t_{ij}$  is explained (i.e.,  $y_{ij} = 1$ ) given its context:

$$P(y_{ij} = \textit{explain} \mid \mathbf{x}_{ij}; \mathbf{w}) = \frac{1}{1 + \exp(-\mathbf{w}^T \mathbf{x}_{ij})}$$

Because we assume that  $\{y_{ij}\}$  are generally not available or are scarce, we introduce the distributional constraint on  $\{y_{ij}\}$  that allows us to infer them simultaneously. Given a sequence of  $N$  units (e.g., chapters)  $j = 1 \dots N$  in the textbook where the term  $t_i$  occurs, we can express the likelihood of a particular sequence of labels  $\{y_{ij}\}_{j=1\dots N}$  across the span of the entire book as follows:

$$P(\{y_{ij}\}_{j=1\dots N} \mid \{\mathbf{x}_{ij}\}_{j=1\dots N}, z_i; \mathbf{w}) = \prod_{j=1}^N P(y_{ij} = \delta[z_i = j] \mid \mathbf{x}_{ij}; \mathbf{w})$$

where  $z_i \in \{1 \dots N\}$  is a categorical variable indicating where in the sequence of units term  $t_i$  is explained, and  $\delta[z_i = j]$  is an indicator function that is equal to 1 (i.e.,  $y_{ij} = \textit{explain}$ ) when the value of  $z_i$  is equal to the unit index  $j$ . Expressed another way, the above likelihood can be viewed as a conditional mixture model where the posterior over  $z_i$  is a distribution over the location in the textbook where the term is explained. Most importantly, the form of the above likelihood naturally encodes our prior belief that a concept is typically explained in one location in a textbook. Doing inference in this model is straight-forward with the

Expectation Maximization algorithm. Additionally, we can easily incorporate labels if we have any available.

## Results

We are now interested in the effectiveness of the distributional constraint introduced in Section 10.3.1 as a way to go around the task of collecting a large amount of annotated data. We first use Chris Bishop's *Pattern Recognition and Machine Learning* textbook for evaluating the effectiveness of this technique, as this textbook is fairly unique in that it provides an index of terms that is annotated with the location of where the term is explained. We note that this data cannot be reliably used as a substitute for manual annotation, as there are no annotation guidelines available. However, this annotation fits well into the structure of our semi-supervised approach, as our hidden variables  $z_i$  are precisely the location in the sequence of units where a particular term  $i$  is explained. To evaluate the effectiveness of the proposed distributional constraint, we control the amount of annotated data that is used during training from 5% of the labeled examples to the complete dataset. Figure 10.2 illustrates the results for three models: (i) a semi-supervised approach, (ii) a fully supervised approach and (iii) a baseline of deterministically assigning the *explain* label to the first occurrence of the term in a textbook. We observe that the semi-supervised approach outperforms the fully-supervised dramatically in cases with almost no training data. We also observe that the effect of additional training data is fairly small, indicating that the simple distributional constraint on the labels based on the structure of the textbook is a powerful enough constraint to induce the labels without explicit annotation. Additionally, we apply this semi-supervised training to 5 additional

textbooks obtained from *OpenStax*, and illustrate the ability to perform aspect classification fairly well across these different domains (Figure 10.3).

We illustrate two interesting posterior distributions over  $z_i$  (location in the textbook where term  $i$  is explained) for two terms in Figure 10.4 and Figure 10.5 (*conditional independence* and *cross validation* respectively) (many of the distributions are unimodal and less interesting for analysis). The  $x$ -axis in both figures corresponds to the linear ordering of the units within the textbook (left end-point corresponds to the beginning of the textbook), and blue stems reflect the probability that term  $i$  is explained in a given unit (from the posterior distribution over  $z_i$ ). The red stems corresponds to the unit where the term is annotated to be explained (i.e., ground truth). In both figures, observe that the term is not explained in its first appearance in the textbook, where the model correspondingly assigns low probability. Although *conditional independence* is given a cursory introduction in the first unit (where the posterior assigns a significant portion of the probability mass), the model correctly assigns the greatest probability mass in the chapter on graphical models, where the concept of conditional independence is explained thoroughly. In Figure 10.5, the model assigns approximately equal probability to the location of explanation for term *cross validation*, though the highest posterior estimate does not agree with the ground truth annotation (the concept of cross validation is explained in the first unit). The model assigns a significant probability mass to the chapter on *support vector machines* where the cross-validation is referred to extensively, but not explained for the first time.

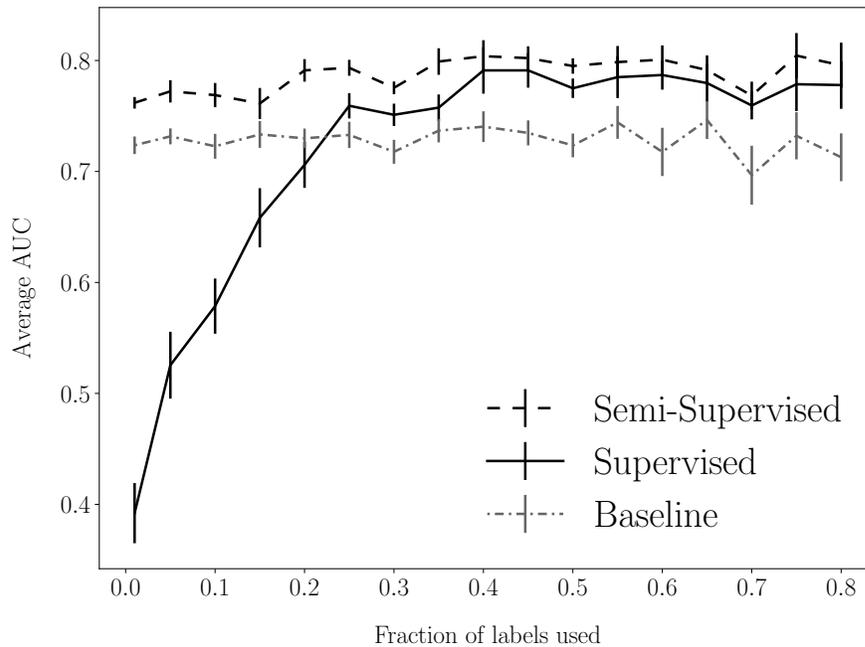


Figure 10.2: Average AUC using 10 fold cross-validation at the task of binary aspect classification, applied to Chris Bishop’s *Pattern Recognition and Machine Learning* textbook. A semi-supervised approach with less than 5% of the labeled examples performs comparably to the fully supervised model with hundreds of labeled examples

## 10.4 Optimal learning paths

Consider now that we have a large collection of documents (e.g., tutorials, papers, textbook chapters). Each such document explains some concepts but also assumes the reader’s knowledge of other concepts (e.g., a tutorial may explain the concept of *normal distribution*, but may assume the knowledge of *probability* and *distribution*). We will now consider that we can reliably classify each term in each document into either the *Explained* or *Assumed* category. Consider that we also have a user who is interested in understanding a specific (target) document (or a set of target documents). The goal is to give a user a self-contained sequence of documents of minimal length that explains all of the concepts needed

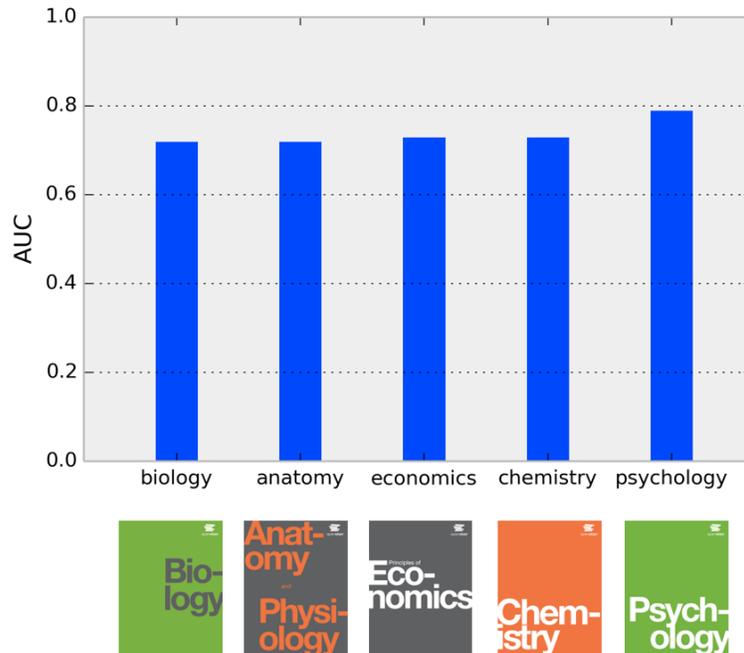


Figure 10.3: Average AUC using 10 fold cross-validation at the task of binary aspect classification, applied to five textbooks. Each model was trained independently for each textbook.

to understand the target document.

Formally each document  $d_i$  in our collection is a set of two sets of terms: the explained terms  $E_i = E(d_i)$  and the assumed terms  $A_i = A(d_i)$ . A term in any document is either explained or assumed, but not both, i.e.,  $A_i \cap E_i = \emptyset$ . We say that the document  $d_i$  is *covered* by a prerequisite set of documents  $P_i$  when:

$$A_i \subseteq \bigcup_{d_j \in P_i} E(d_j)$$

In other words the document is covered when every one of its assumed terms is explained by at least one document in the prerequisite set. For any prerequisite

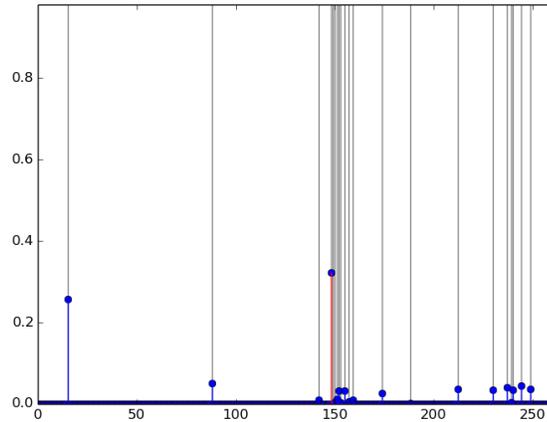


Figure 10.4: Posterior distribution over the units explaining *conditional independence* inside Chris Bishop’s “Pattern Recognition and Machine Learning” textbook (left is the beginning of the textbook, right is the end of the textbook). Red color the location where the index indicates the term is actually explained (gold standard annotation).

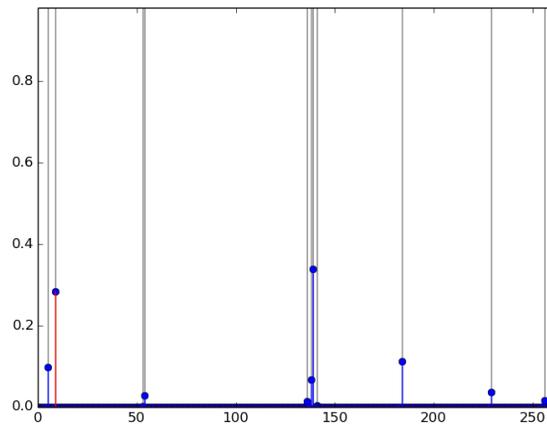


Figure 10.5: Posterior distribution over the units explaining *cross validation* inside Chris Bishop’s “Pattern Recognition and Machine Learning” textbook (left is the beginning of the textbook, right is the end of the textbook). Red color the location where the index indicates the term is actually explained (gold standard annotation).

set that covers this document, the documents in the prerequisite set need to be covered as well, recursively until all documents have been covered. We assume the existence of documents with no prerequisites (leaves), i.e., those documents for which  $A. = \emptyset$ . The goal is to find a smallest *self-contained* set of documents  $P$ , i.e., a set of documents such that all the documents in  $P$  are covered and  $d_0 \in P$ , where  $d_0 = \{A_0, E_0\}$  is the target document of interest to the user. Figure 10.7 illustrates a feasible solution to an example problem. Without additional restrictions, solutions to this problem can contain cyclical dependencies. Such cycles don't make sense in our setting. Thus an important restriction is that the the set of documents  $P$  can be ordered such that every document in the sequence is covered by the preceding documents in the sequence. Let  $\mathbf{p}$  be a sequence of documents of length  $K$ , where  $\mathbf{p}_k$  is the  $k^{th}$  document in the sequence, then we seek:

$$\begin{aligned}
 & \text{minimize } |\mathbf{p}| \\
 & \text{s.t. } \forall k : A(\mathbf{p}_k) \subseteq \bigcup_{k'=0}^{k-1} E(\mathbf{p}_{k'}) \\
 & d_0 \in \mathbf{p}
 \end{aligned} \tag{10.1}$$

## 10.5 Problem hardness

### Set Cover Problem (SC)

- input: universe  $U$ , and sets  $S_1, S_2, \dots, S_m \subseteq U$
- output:  $I \in [m]$ , so that  $\bigcup_{i \in I} S_i = U$

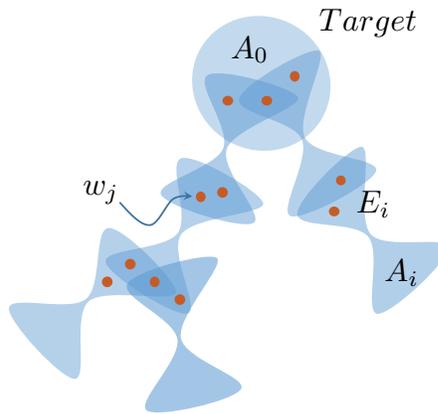


Figure 10.6: Each document is represented by a blue shaded region: the top part corresponds to the explained set  $E_i$  and the bottom part corresponds to the assumed set  $A_i$ . Red dots correspond to terms. This is an example of a feasible solution, where each document is *covered*.

- goal: minimize  $|I|$ .

### Learning Paths Problem (LPP)

- input: set of documents  $D$ , goal document  $g \in D$ , set of concepts (terms)  $C$ ; each document has a set of explained terms and assumed terms (formally set functions  $A : D \rightarrow 2^C$  and  $E : D \rightarrow 2^C$ )
- output: a sequence of documents  $d_1, d_2, d_3, \dots, d_k, d_{k+1} = g$ , so that  $A(d_1) = \emptyset, A(d_{i+1}) \subseteq \bigcup_{j=1}^i E(d_j)$  for  $i = 1, 2, \dots, k$
- goal: minimize  $k$  (the number of documents excluding  $g$ ).

### Reduction to show LPP is at least as hard as SC

**Lemma 1.** *If there exists an  $\alpha$ -approximation for LPP, then there exists an  $\alpha$ -approximation for SC.*

*Proof.* Given an instance of SC, create the following (layered) LPP instance: Let  $D = \{0, 1, 2, \dots, m\}$ ,  $g = 0$ , and  $C = U \cup \{x\}$ . Define  $E(0) = \emptyset$  and  $A(0) = U$ , and  $E(i) = S_i$  and  $A(i) = \emptyset$  for  $i = 1, 2, \dots, m$ .

**Claim 1:** Any feasible solution to the constructed LPP instance of size  $k$  (with no documents repeated) corresponds to a set cover of size  $k$ . *Proof:* Let  $d_1, d_2, \dots, d_k, d_{k+1} = 0$  be a feasible solution to the LPP instance. Because it is a feasible solution, this means that  $A(0) = A(d_{k+1}) \subseteq \bigcup_{j=1}^k E(d_j) = \bigcup_{j=1}^k S_{d_j}$ , and  $A(0)$  was defined to be  $U$ . In other words,  $\bigcup_{j=1}^k \{d_j\}$  is a set cover, and it has size  $k$ .

**Claim 2:** Any set cover of size  $k$  corresponds to a feasible solution to the constructed LPP instance of size  $k$ . *Proof:* Let  $I = \{i_1, i_2, \dots, i_k\}$  be the set cover of size  $k$ . Then the sequence  $i_1, i_2, \dots, i_k, 0$  is a feasible solution to the LPP instance, since  $A(i_j) = \emptyset$  for all  $i_j \in I$ , and  $\bigcup_{j=1}^k E(i_j) = \bigcup_{j=1}^k S_{i_j} = U$  because  $I$  is a set cover, and the length of the solution is  $k$ .

This is a polynomial time reduction, so if there exists an  $\alpha$ -approximation for LPP, then we can get  $\alpha$ -approximation for SC by using this reduction.  $\square$

**Corollary 3.** *LPP is NP-hard.*

**Corollary 4.** *Unless  $NP \subseteq TIME(n^{O(\log \log n)})$ , there is no  $\epsilon > 0$  so that there is a polynomial time algorithm to approximate layered LPP within a factor of  $(1 - \epsilon) \ln n$ .*

*Proof.* Feige (1998) shows this for set cover, and the reduction shows that this is therefore also true for LPP.  $\square$

## Stronger Reduction

The reduction above can be replicated for each document that is not  $g$ , and we get a layered stronger reduction. The intuition is that we get multiple disjoint copies of the set cover instance. As an example, consider extending the reduction above to a next layer. We create disjoint copies of the set cover instance for each document in the original reduction, as follows. We expand  $C$  with  $m$  unique copies of  $U$ , one for each document  $i = 1, 2, \dots, m$ . We denote these copies by  $U^{(i)}$ , and its elements denoted by  $e^{(i)}$ , i.e.,  $U^{(i)} := \{e^{(i)} : e \in U\}$  for  $i = 1, 2, \dots, m$ . The assumed terms of document  $i$  are now set to  $U^{(i)}$  for  $i = 1, 2, \dots, m$ . We expand  $D$  with  $m$  unique copies of all sets in the set cover instance. We denote these new documents by  $d_j^{(i)}$ , for each document  $i \neq 0, j = 1, 2, \dots, m$ . We set  $E(d_j^{(i)}) = S_j^{(i)} := \{e^{(i)} : e \in S_j\}$ , and  $A(d_j^{(i)}) = \emptyset$ . Documents  $0, 1, \dots, m$  (the documents that were already present in the first reduction) will be referred to as the “original documents”.

We now have the following generalizations of the claims above.

Claim 2 (generalized): Any set cover of size  $k$  corresponds to a feasible solution to the constructed LPP instance of size  $k^2 + k$ . Proof: For each original document there is a corresponding LPP instance that is exactly the same as the LPP instance of the original reduction. Therefore, the solution as described above can be used for every original document, giving a path of length  $k$  for each of these documents. Since there are  $k$  original documents in the LPP solution, and each of these documents requires  $k$  documents, the total number of documents in the overall LPP solution is  $k^2 + k$ .

Claim 1 (generalized): For any feasible solution to the constructed LPP in-

stance of size  $K$  (with no documents repeated) there exists a set cover of size at most  $\sqrt{K}$ . Consider the set cover solutions associated with each of the original documents in the LPP solution. If all of the solutions are larger than  $\sqrt{K}$ , then the solution of the LPP solution is larger than  $\sqrt{K} \times \sqrt{K} + \sqrt{K} = K + \sqrt{K}$ , a contradiction. Therefore there must be at least one solution of size at most  $\sqrt{K}$ .

We can repeat this procedure as often as we like. Suppose we repeat the procedure  $L$  times (where each time we do the procedure on *all* leaf documents).

Claim 2 (further generalized): Any set cover of size  $k$  corresponds to a feasible solution to the constructed LPP instance of size  $k^L + k^{L-1} + \dots + k^2 + k$ .

Claim 1 (further generalized): For any feasible solution to the constructed LPP instance of size  $K$  (with no documents repeated) there exists a set cover of size at most  $\sqrt[L]{K}$ .

**Corollary 5.** *For every  $L \geq 1$ , unless  $NP \subseteq TIME(n^{O(\log \log n)})$ , there is no  $\epsilon > 0$  so that there is a polynomial time algorithm to approximate layered LPP within a factor of  $(1 - \epsilon) \ln^L n$ .*

## 10.6 ILP formulation

We formulate an Integer Linear Program (ILP) that finds a minimum length self-contained sequence  $\mathbf{p}$  of at most  $K$  documents such that it covers a user's document of interest  $d_0$ . Consider that we have a total of  $D$  documents. We define the following variables:

$$x_i^k \in \{0, 1\} \quad \text{document } d_i \text{ is in } k^{\text{th}} \text{ position in the sequence}$$

We define the following constants:

$e_{ij} \in \{0, 1\}$  Term  $j$  is explained in document  $i$

$a_{ij} \in \{0, 1\}$  Term  $j$  is assumed in document  $i$

Each assumed term in a document in position  $k$  must be explained by at least one document up to (but not including) the document in position  $k$ . This can be expressed via the following constraint:

$$\sum_{k'}^{k-1} \sum_i^D e_{ij} x_i^{k'} \geq \sum_i^D a_{ij} x_i^k \quad \forall j \forall k$$

Each position in the sequence contains at most 1 document:

$$\sum_i^D x_i^k \leq 1 \quad \forall k$$

User's preference of covering a document of interest  $d_0$  is an additional constraint:

$$\sum_k^K x_0^k = 1 \quad \forall k$$

Finally, the objective is to minimize the number of documents in the sequence:

$$\text{minimize } \sum_k^K \sum_i^D x_i^k$$

The above formulation also allows us to directly incorporate the user's prior knowledge into this optimization problem. If we represent a user as a set of *explained* terms, i.e., terms that the user is assumed to have mastered, then the constraints corresponding to these terms may simply be dropped from the formulation.

In the most general case, this formulation has  $D^2$  variables and  $O(D^2 \times V)$  constraints, where  $V$  is the number of terms in the vocabulary. In practice,

however, we will often limit the maximum allowable sequence length to a fairly small constant (e.g., 10, as done in our experiments), reducing the order of the problem to  $O(D)$  variables and  $O(D \times V)$  constraints.

While in extremely large settings (hundreds of thousands of documents), even with a small  $K$ , solving this ILP directly is infeasible, in practice, we find that that we can obtain exact solutions using LP relaxation and a vanilla Branch and Bound (using GLPK<sup>1</sup>) within several seconds, even with a many as 1,000 documents and hundreds of terms. Developing an approximation algorithm based on rounding the LP solution is future work.

## 10.7 Variation: Layered Set-Cover

Consider the following variation on the problem: given the target document  $d_0$ , the user is given a tree of documents, with  $d_0$  at the root. Children nodes in the tree correspond to the set of documents that cover all of the assumptions of the parent. The goal is to find the tree with the minimum number of nodes (documents), such that every document in the tree is covered. This problem is different from the previous formulation in that in this formulation the children (immediate descendants) are required to cover the parent document completely. In the former formulation, any descendant (regardless of depth) can participate in the cover. As a result, the former formulation could result in fewer nodes, since the same descendant can participate in covering multiple ancestors, regardless of the distance between them. The latter formulation would “double count” the same document in multiple occurrences in the tree. The advantage of this

---

<sup>1</sup><https://www.gnu.org/software/glpk/>

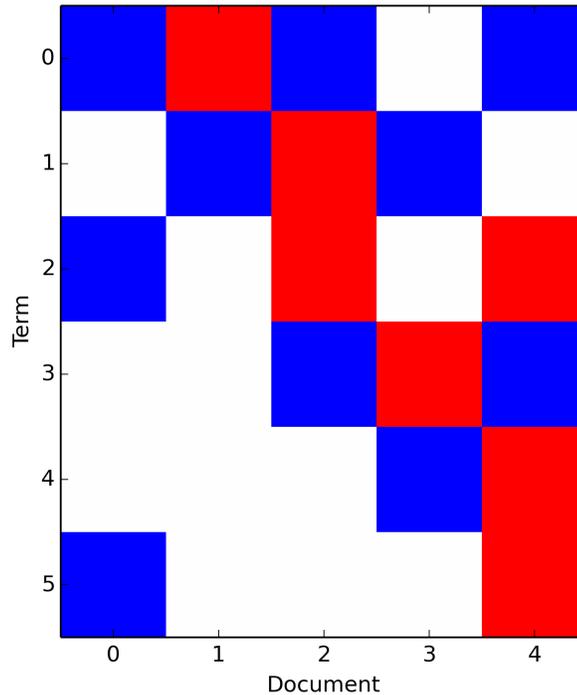


Figure 10.7: Optimal solution for a small toy problem obtained with an ILP. Each row is a unique term and each column is a document corresponding to a position in a sequence (from left to right). A blue square indicates that the term is explained in the document, and a red square that the term is assumed. Notice that all of the constraints are satisfied: each term is explained in some document before it is assumed and each position in the sequence is occupied by a single document.

formulation, however, is that it gives a natural Dynamic Programming algorithm with an approximation guarantee. This formulation also can be justified in the following way: even if some prerequisite document that appeared far down in the tree satisfies part of the prerequisites of the current document, the user might nevertheless benefit from seeing that document again, e.g., to remind them of the necessary concepts.

We call this problem `LAYEREDSETCOVER`. Formally, we are given a universe

of documents  $\{d_i\}$ , where as before, each document is a set of two sets  $d_i = \{A_i, E_i\}$ , where  $E_i$  and  $A_i$  are the sets of *assumed* and *explained* terms respectively. Given a maximum depth  $K$ , find a tree with the minimal number of nodes (documents), such that the root of the tree is the user's target document  $d_0$  and each node is *covered* (defined next). An edge  $d_i^{k-1} \rightarrow d_j^k$  (edge between document  $d_i$  at depth  $k - 1$  to document  $d_j$  at depth  $k$ ) implies that  $E(d_j) \cap A(d_i) \neq \emptyset$ , i.e., document  $d_j$  explains at least one assumed concept in document  $d_i$ . A node is considered *covered* when

$$A(d_i) \subseteq \bigcup_{d_j \in \text{Child}(d_i)} E(d_j)$$

i.e., when the child documents explain all of the assumptions of the parent document. For the problem to have a feasible solution, there must exist documents in the set that have an empty assumed set, i.e.,  $A_i = \emptyset$ . Similar to the first problem, by reduction from SETCOVER, we can show that LAYEREDSETCOVER is also NP hard.

## Approximation algorithm

Assign a weight  $w_i^k$  to each document at each layer of the graph. Let  $w_i^k$  be the total number of documents needed to understand document  $d_i$  at layer  $k$ , i.e., the total number of descendants of node  $d_i$ . For any node  $d_i$ , the weight  $w_i^k$  can be expressed recursively in terms of the weights of its children:

$$w_i^k = \sum_{j \in \text{child}(d_i^k)} (w_j^{k+1} + 1)$$

i.e., the number of descendants of node  $d_i^k$  is the number of children of  $d_i^k$  plus the number of descendants of each child (in our convention, the index of a child

is greater than the index of the parent). Let  $w_i^k = 0$  for all documents with empty assume set, i.e.,  $A_i = \emptyset$  (since these documents do not need to be covered, the number of its descendants is naturally zero). The optimization objective of minimizing the total number of nodes in the tree can be expressed in an equivalent way by requiring to minimize the weight of the root document  $d_0$ :

$$\min w_0^0$$

since  $w_0^0$  counts the total number of documents (nodes) in the graph. At the root-level, the above problem is the standard MINCOSTCOVER, a weighted set-cover problem, where the objective is to find a minimum cost cover of the root document, but where the costs of each set are unknown. We can, however, express the weighted set-cover recursively in terms of the smaller partial weighted set-covers:

$$\text{LAYEREDCOVER}(d_i, k) = \text{MINCOSTCOVER} \left( \begin{cases} 1 + \min_{k' > k} \text{LAYEREDCOVER}(d_0, k') \\ 1 + \min_{k' > k} \text{LAYEREDCOVER}(d_1, k') \\ \vdots \\ 1 + \min_{k' > k} \text{LAYEREDCOVER}(d_N, k') \end{cases} \right)$$

where for illustration,  $\text{LAYEREDCOVER}(d_i, k)$ , is the minimum weight tree rooted at  $d_i^k$  (i.e., node  $i$  at level  $k$ ) and  $\text{MINCOSTCOVER}$  is the weighted set cover, shown explicitly in terms of the weights of the individual elements over which the minimum set cover is computed ( $N$  is the total number of documents). Clearly the above can be solved using Dynamic Programming, where the partial solutions of the form  $\text{LAYEREDCOVER}(d_i, k)$  can be computed once and saved into a table of size  $N \times K$  (number of documents by number of layers in the graph).

The table can be computed bottom-up, by starting at layer  $k = K$  and ini-

tializing all documents with  $A_i = \emptyset$  (i.e., no assumptions) with weight  $w_i^K = 0$  (the remaining documents at that layer are left un-initialized, or with  $w_i^K = \infty$ ). At the next level  $k = K - 1$ , compute the MINCOSTCOVER for every document, in terms of only the documents at the level below whose weight has been already assigned (i.e.,  $w_i^{k-1} \neq \infty$ . At level  $k = K$ , these are only the documents with weight 0). Proceed in that way for every layer until  $k = 0$ . At each layer, MINCOSTCOVER is computed for *every* document, including those whose weight has been computed at a layer below. If the weight of the new MINCOSTCOVER is lower than the weight of the cover of the same document at a layer below, the new weight is saved in the table, otherwise the weight of the previous layer is used. Let  $cost(d_i, k)$  be the entry in the table corresponding to the optimal solution to LAYEREDCOVER( $d_i, k$ ), then:

$$cost(d_i, k) = \min \left( cost(d_i, k + 1), \text{MINCOSTCOVER} \left( \begin{array}{c} 1 + cost(d_0, k + 1) \\ 1 + cost(d_1, k + 1) \\ \vdots \\ 1 + cost(d_N, k + 1) \end{array} \right) \right)$$

At the layer  $k = 0$  (root), the cost of the target document  $d_0$  obtained in this way is the optimal solution. The edges in the graph can be reconstructed by retracing the set-covers at every layer to  $k = K$ .

**Lemma 2.** *The approximation ratio of the dynamic programming algorithm is  $O(\log^K n)$  where  $n$  is the maximum number of assumed terms in any document and  $K$  is the number of layers in the graph.*

*Proof.* The approximation ratio of MINCOSTCOVER is  $\mathcal{O}(\log n)$ , where  $n$  is the number of elements to be covered (in our setting, elements are *terms* to be covered). At every layer in the graph, starting from the bottom, the approximation gets worse, since the weights of the elements at every layer themselves

become approximations of the true weights (since at each layer, the weight of each document is itself a solution to a MINCOSTCOVER, which we can only solve approximately in polynomial time).

Consider the cost of covering the documents at layer  $k = K - 1$ . These documents can be covered only in terms of the documents at layer  $k = K$ , which include only the “cost 0” documents, i.e., those that have no prerequisites (their assumed sets are empty). The cost of covering a document at layer  $k = K - 1$  using the greedy approximation algorithm for MINSETCOVER is then:

$$\begin{aligned} \text{cost}(d_i, K - 1) &\leq O(\log n_i) \text{cost}(\text{MINCOSTCOVER}(d_i, K - 1)) \\ &= O(\log n_i) \text{OPT}(d_i, K - 1) \end{aligned}$$

where  $\text{cost}(\text{MINCOSTCOVER}(d_i, k))$  is the cost of the minimal weighted set cover using the document weights in layer  $k$ . Because the cover at level  $K - 1$  can be made only with documents whose weight is known exactly, the above is also equal to  $\text{OPT}(d_i, K - 1)$ , i.e., the minimum cost of covering document  $d_i$  at layer  $K - 1$ .  $n_i$  is the number of *assumed* terms in document  $d_i$ . Consider now the cost of covering the document  $d_i$  at layer  $k = K - 2$ . This cost can be expressed as follows:

$$\begin{aligned} \text{cost}(d_i, K - 2) &\leq O(\log n_i) \text{cost}(\text{MINSETCOVER}(d_i, K - 2)) \\ &= O(\log n_i) \min(\{\text{cost}(C_1, K - 1), \dots, \text{cost}(C_m, K - 1)\}) \end{aligned}$$

where  $\{C_1, \dots, C_m\}$  is the set of all covers (subsets that cover all elements of  $A(d_i)$ ), i.e.,

$$A(d_i) \subseteq \bigcup_{d_j \in C_i} E(d_j)$$

where  $\text{cost}(C_i, K - 1)$  is the cost of the cover  $C_i$  at layer  $K - 1$ . But  $\text{cost}(C_i, K - 1)$  is the number of documents in the cover, plus the costs to cover each document

in that cover:

$$\text{cost}(C_i, K - 1) = \sum_{d_j \in C_i} 1 + \text{cost}(d_j, K - 1)$$

But since the cost of each document in the layer below is not known exactly, we can only obtain the upper bound on  $\text{cost}(C_i, K - 1)$  in terms of the upper bounds on the weight of each document:

$$\text{cost}(C_i, K - 1) \leq \sum_{d_j \in C_i} 1 + O(\log n_j) \text{OPT}(d_j, K - 1)$$

where  $n_j$  is the number of assumed terms in document  $j$ , i.e.,  $|A(d_j)|$ . Let  $n$  be the maximum number of assumed terms in any document, i.e.,  $n = \max\{n_j\} \forall j$ :

$$\begin{aligned} \text{cost}(C_i, K - 1) &\leq \sum_{d_j \in C_i} 1 + O(\log n) \text{OPT}(d_j, K - 1) \\ &\leq O(\log n) \sum_{d_j \in C_i} 1 + \text{OPT}(d_j, K - 1) \\ &= O(\log n) \text{OPT}(C_i, K - 1) \end{aligned}$$

where  $\text{OPT}(C_i, K - 1)$  is the true cost of cover  $C_i$ . Therefore we have:

$$\begin{aligned} \text{cost}(d_i, K - 2) &\leq O(\log n_i) \min(\{\text{cost}(C_1, K - 1), \dots, \text{cost}(C_m, K - 1)\}) \\ &\leq O(\log n)^2 \min(\{\text{OPT}(C_1, K - 1), \dots, \text{OPT}(C_m, K - 1)\}) \\ &= O(\log n)^2 \text{OPT}(d_i, K - 2) \end{aligned}$$

By induction, we obtain:

$$\text{cost}(d_i, k = 0) \leq O(\log^K n) \text{OPT}(d_i, k = 0)$$

i.e., at the root ( $k = 0$ ), the upper-bound of the the weighted set cover approximation deteriorates exponentially with the depth of the graph (number of layers). □

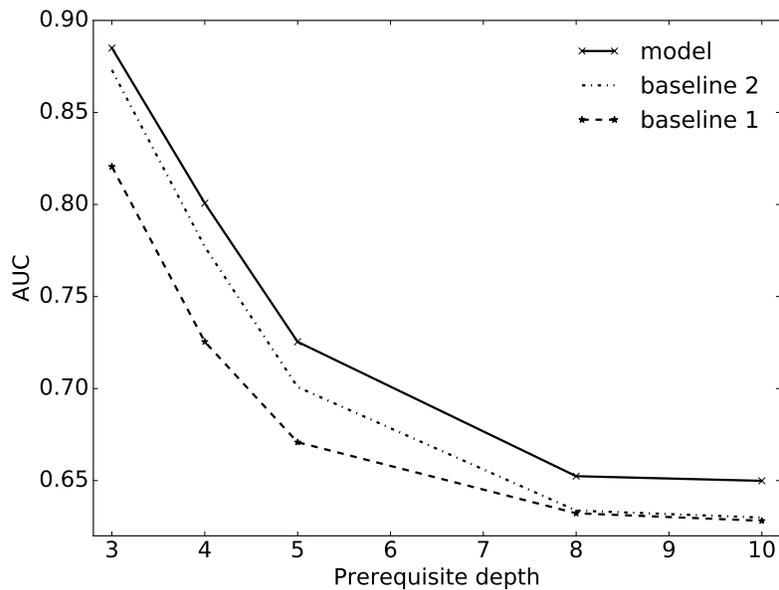


Figure 10.8: Term aspect classification is useful at the task of recovering prerequisites for units within a textbook. The  $y$ -axis is the average AUC at the task of predicting whether a particular unit is a prerequisite of another unit, based on three metrics. The metric that incorporates the *Explain/Assume* classifier performs best (solid line).

## 10.8 Experiments

### 10.8.1 Prerequisites

In order to evaluate the *Explain/Assume* classifier in an end-to-end setting, we employ the output of this classifier in the task of predicting prerequisites in a dataset where the prerequisites have been explicitly annotated. One such resource is *Rice University's Online Statistics Textbook*, which in addition to the text content, provides an explicit dependency graph annotating prerequisite relations between pairs of units (units are at the level of chapter sections). We propose a metric for scoring a pair of units according to their prerequisite relationship

based only on the terminology of both units and the output of the *Explain/Assume* classifier. The proposed “prerequisite score” is defined as follows:

$$P(d_a \rightarrow d_b) = \frac{\sum_{t_i \in d_b} n_i^a \mathbb{1}[t_i \text{ assumed in } d_b \wedge \text{ explained in } d_a]}{\sum_{t_j \in d_a} n_j^a \mathbb{1}[t_j \text{ explained in } d_a]}$$

where  $n_i^a$  is the number of occurrences of term  $i$  in document  $d_a$ . Since the above score is guaranteed to be in the  $[0, 1]$  range, we can interpret it as a probability  $P(d_a \rightarrow d_b)$ , a probability that document  $a$  is a prerequisite of document  $b$ . There is an intuitive interpretation to the above score: a document can be considered a strong prerequisite of a target document when it explains all of the assumed terms in the target document and nothing more. We can convince ourselves that in this case the score as defined above will be equal to 1. A document that explains too many unrelated concepts will suffer a penalty with respect to its prerequisite score to another document. Furthermore, we consider the relative frequency of the explained term in the prerequisite document as an additional signal of that term’s importance. We find that this additional information increases the performance of prerequisite classification (discussed at the end of this section).

Because the output of the *Explain/Assume* classifier is a probability, rather than a class, we can relax the above score to directly incorporate the uncertainty in the classification:

$$P(d_a \rightarrow d_b) = \sum_{t_i \in d_b} \frac{n_i^a P(t_i \text{ explained in } d_a)}{\sum_{t_j \in d_a} n_j^a P(t_j \text{ explained in } d_a)} \quad (10.2)$$

Note that in addition to relaxing the requirement of an explicit *Explain* or *Assume* label, we also drop the requirement that only the assumed terms need to be explained to count towards the prerequisite score. This distinction is optional, but it encodes an important assumption on the kinds of “prerequisites” that this score will discover. This also brings up the importance of being precise about the definition of a prerequisite. A document  $a$  is a strict prerequisite of

document  $b$ , if document  $a$  explains a subset of the assumptions in document  $b$ . However, we can relax this definition by *not* requiring that the terms explained in the prerequisite ( $a$ ) are strictly assumed in the target ( $b$ ). In other words, a document that explains a subset of the terms also explained in the target and *nothing else*, will have a score of 1 according to the above equation. In practice this corresponds to documents that explain the same concepts but in a simpler way (since they explain only a subset of the explained concepts in the target), and this is often a desired behavior in a learning sequence. For example, before reading a more advanced article on *Support Vector Machines*, the learner might want to read a more basic introduction to *Support Vector Machines*, although from the perspective of term classifications, both documents explain the same concept.

### **Reconstructing prerequisites**

*Rice University's Online Statistics Textbook* provides a valuable resource for evaluating the effectiveness of the *Explain/Assume* classification at the task of predicting prerequisite relations between documents. The textbook consists of 112 units at the granularity of chapter sections, annotated as a directed graph, i.e., specifying a directed edge between a pair of units if one unit is considered a prerequisite of another unit. We process the raw HTML files of the textbook by removing markup, segmenting sentences and extracting terminology (obtained from the index) features as described in Section 10.3. We pose the problem of prerequisite relation prediction as a standard binary classification task, i.e., predicting for each pair of units in the book whether one unit is a prerequisite of another, where we consider a pair of units to be in a gold-standard prerequisite relation if there is a directed path between them in the graph. AUC is a convenient metric for

evaluating performance in this prediction task, as the output of our scoring metric (Equation 10.2) is already scaled between 0 and 1. Note that the model trained only on the PRML corpus was used for term-aspect classification in this task. Figure 10.8 illustrates the results for three different models, as a function of the prerequisite depth, i.e., stratifying the classification results for a pair of units by the maximum distance between them in the graph. The three models evaluated are as follows:

- **Model** Prerequisite score is computed with (10.2).
- **Baseline 1** Prerequisite score is computed with (10.2), but with all  $n_i^a$ ,  $n_j^a$  and  $P(t. \text{ explained in } \cdot)$  set to 1. This baseline is equivalent to a ratio between the number of overlapping terms between a pair of documents and the number of terms in the prerequisite, i.e.,  $\frac{|d_a \cap d_b|}{|d_a|}$ .
- **Baseline 2** Prerequisite score is computed with (10.2), but with  $P(t. \text{ explained in } \cdot)$  set to 1.

Each baseline illustrates the effect of *not* including a component of the scoring function in (10.2). Our first conclusion from the results in Figure 10.8 is that the output of the *Explain/Assume* classifier provides an important signal in predicting the prerequisite relationship between documents. Furthermore, the relative frequency of the explained terms in the prerequisite document provides an additional gain in performance. This can be explained by Figure 10.1(b): the performance of the *Explain/Assume* classifier is greater in the higher term-frequency regime; discounting low-frequency terms (that are also likely less important to the content) reduces the classification noise and boosts the performance at the prerequisite prediction task. An additional observation is that the performance of the pairwise prerequisite classification improves for pairs of units that are closer

in the graph, i.e., with less units in between. This is easily explained: units that are farther apart typically share less terminology, making the estimates based on terminology overlap noisier.

It is also interesting to note that the simplest baseline that considers only the ratio of overlapping terms between a pair of documents to the total number of terms in the prerequisite document does surprisingly well, especially well for pairs of documents closer together. This can be explained as follows: in a sequence of units like those in a textbook, units that are prerequisites tend to be less advanced, i.e., have less terminology, since less of it was introduced up to that point. Thus, units that are prerequisites, at least in a textbook, would be fairly predictable from the relative frequency of overlapped terms alone.

### **10.8.2 Scaling to the web**

We collect and release two web corpora of educational content in the areas of Machine Learning and Statistics. Both corpora were collected using Bing Search API, by querying for short permutations of terms collected from the index of the *Pattern Recognition and Machine Learning* and *Rice University's Online Statistics Textbook*. The two corpora contain 42,000 and 1,000 documents respectively – a mixture of HTML and PDF files, pre-processed and converted to plain text. The difference in size of the two corpora is due to a smaller set of keywords used in the query set, and used primarily to rapidly validate the proposed model for path optimization. Consequently, because of a smaller term vocabulary, the smaller corpus is significantly less noisy (less irrelevant documents). The union of the terminology from the index of both textbooks was used as the

vocabulary in processing each document. Additionally, terminology variations and abbreviations were consolidated using the link data from Wikipedia, e.g., terms *EM*, *E-M*, *Expectation-Maximization*, are all mapped to the same concept of *EM* in the terminology extraction stage.

Following the extraction of terminology from each webpage, each term is classified using the *Explain/Assume* classifier trained on the *Pattern Recognition and Machine Learning* textbook. We train this classifier in a fully supervised setting using all of the annotated data. In the next several sections, we present the analysis of the two web corpora and demonstrate the effectiveness of the proposed approach to connecting educational resources on the web.

### 10.8.3 Diversity of assumptions

The web is a unique setting, that unlike a traditional textbook or a course, offers a multitude of diverse explanations of the same concept. This diversity potentially enables the level of personalization that is not possible in traditional resources. We can analyze the diversity in the educational content on the web by looking at a slice of the web resources that share the same topic, but differ in their underlying assumptions and explanations. Figure 10.9 illustrates two articles that are both on the topic of *Expectation Maximization*. However, the two articles differ significantly in their assumptions on the background of the reader. Article 1 (left in Figure 10.9) is a very basic introduction to the topic and does not assume the knowledge of even the concept of *maximum likelihood*, which under most traditional curricula is assumed to be the prerequisite. Article 2 (right in Figure 10.9), however, assumes the knowledge of many more concepts such as

*posterior probability, likelihood function* and *maximum likelihood*. This difference in the distribution of the underlying assumptions is explained by the fact the Article 1 is a very basic introduction to the topic, intended for an audience not in the area of statistics or machine learning. Article 2, however, is a significantly more thorough and a more technical introduction to the concept of the *Expectation Maximization* algorithm and thus assumes significantly more prerequisite background in the areas of statistics and machine learning. It's important to note that this distinction between the two documents cannot be easily made from their titles, or other surface cues: both documents are approximately the same length and their titles do not give away the level of technical detail. Their text content, however, provides the necessary cues to this information.

#### **10.8.4 Fundamental prerequisites**

We apply the prerequisite metric in Equation 10.2 between a pair of web-pages in the large 42,000 web-page corpus. To illustrate the relationship between major topics in machine learning and statistics, we group web-pages by the search-term that was used to retrieve them, considering the "search-term" node as a super-node that subsumes the incoming and outgoing edges of its constituent documents. Figure 10.12 illustrates the in- and out-degree statistics of this graph. This figure provides a broad visualization of the "web of learning" in the machine learning/statistics domains, illustrating the fundamental prerequisites consisting primarily of probability and statistics topics (e.g., *conditional probability, joint probability, bayes rules*) and the topics that are built on these fundamentals (e.g., *expectation maximization, bayesian regression, graphical models*). On careful observation, the graph also contains some noise. For example *factor analysis*

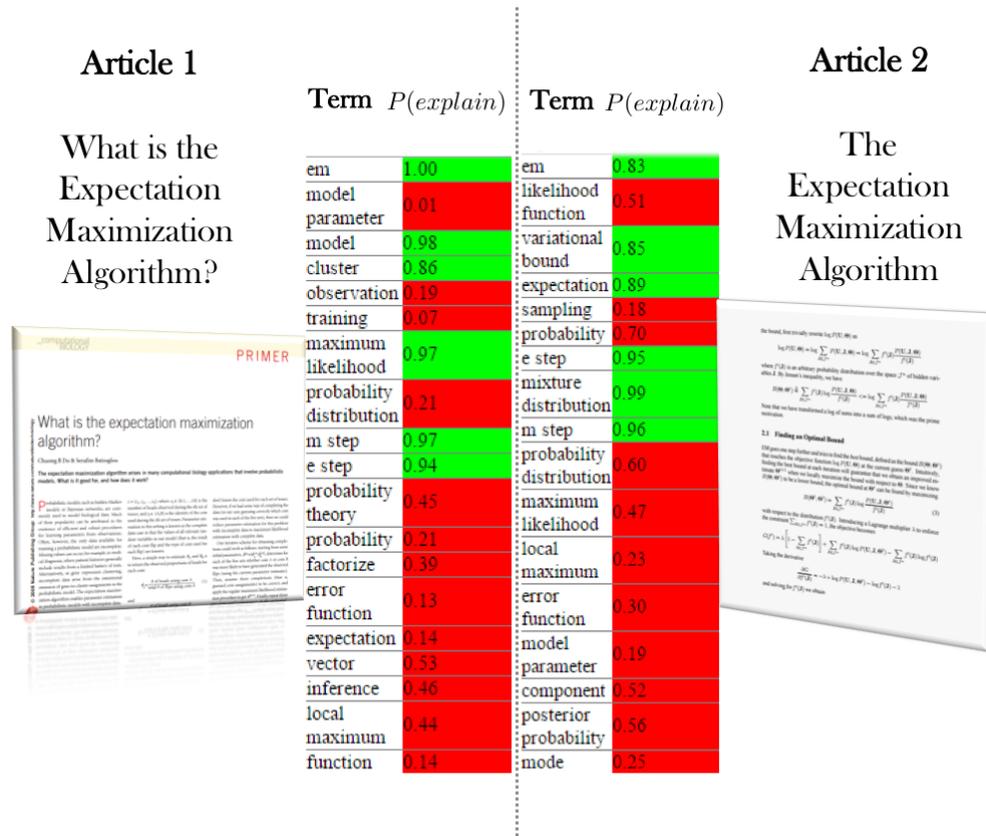


Figure 10.9: An example of two different web-pages about the same topic: *Expectation Maximization*, together with each page’s terminology and its classification into either the *Explained* class (green) or the *Assumed* class (red). Observe that the two pages, while about the same topic, are different in what they assume about the reader. The article on the left is a very basic introduction to this topic, while the article on the right is written for experts.

is typically not a concept that constitutes a prerequisite for many other topics, but appears as such because of the lack of term-sense disambiguation in the pipeline and because the term *factor* is a significantly overloaded term across many sub-domains of machine learning and statistics.

Although the conclusion from this graph is expected, and is something that is typically reflected in classical textbooks and traditional curricula – the power



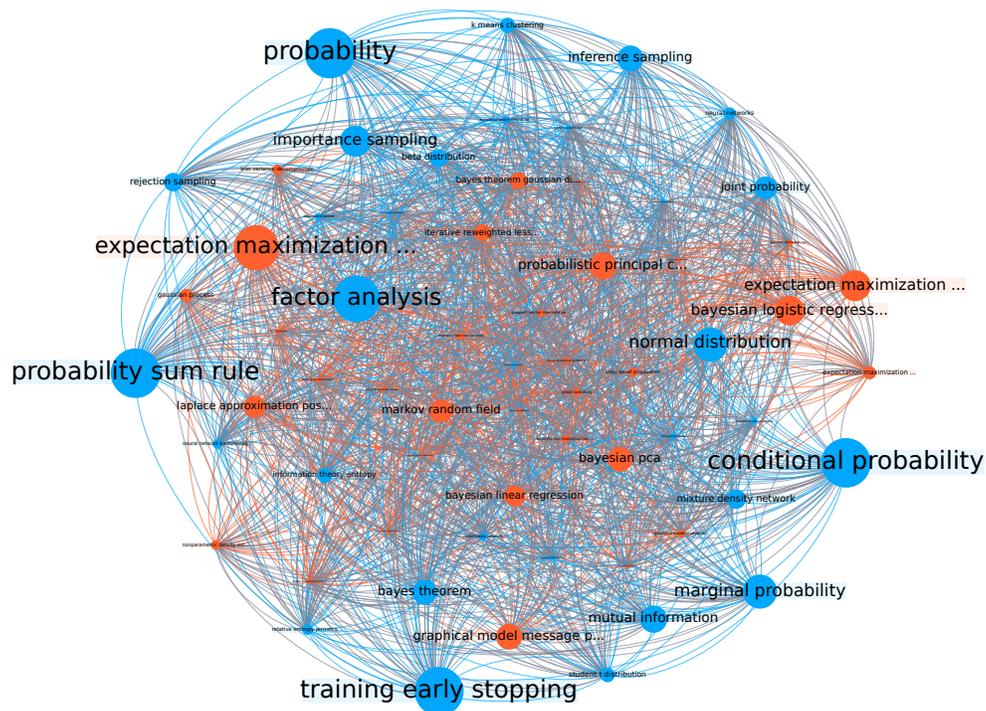


Figure 10.12: A high-level view of a 42,000 document cross-section of the web in the areas of Machine Learning and Statistics. Each node in this graph represents a cluster of web-pages obtained using the same keyword (displayed) to Bing Search API. Fundamental “source nodes” with high out-degree (blue) represent web-pages that explain concepts assumed in many other pages; complex “sink nodes” with high in-degree (orange) represent web-pages that assume concepts explained in many others.

of the technique lies in the flexibility of applying it within and across arbitrary domains, giving the ability to quickly visualize the prerequisite relationships between fundamental concepts in any domain.

Figure 10.10 illustrates the result of optimizing a learning path over the web corpus of 1,000 documents for the target web-page on the topic of “Maximum Likelihood Estimation”. Sequences were optimized using the ILP formulation

described in Section 10.6 using the GLPK Branch and Bound solver. Red rectangles correspond to terms for which the predicted label is *assumed* in the given document, and blue otherwise. In addition to the term-coverage diagram, we also illustrate the prerequisite dependencies extracted from the term coverage data: a directed edge is drawn to a document from the closest prerequisite in the sequence that covers at least one assumed term in the document. In the example in Figure 10.10, the target web-page is a fairly technical article on *Maximum Likelihood Estimation* that assumes the reader's understanding of the concepts such as the *likelihood function* which is pivotal for understanding the concept of *maximum likelihood*. As a consequence, the web-page that is placed immediately before in the optimal sequence are slides which consist of a more basic introduction to the *maximum likelihood*. Furthermore, the original target article assumes the reader's familiarity with *Generalized Linear Models* (which is in fact the previous section of the lecture notes of that series, indicating it as a prerequisite). The resulting sequence also contains an additional prerequisite on this topic. Finally, an interesting observation is that while the target article is fairly advanced in its assumptions about the reader's knowledge of probability, it actually goes into surprising depth in explaining the concept of a *derivative* and maximizing a function using derivatives from scratch, which is another important prerequisite to the concept of *maximum likelihood*. This is highly unconventional in traditional textbook and course curricula. This again underlines the advantage of working with the assumptions at document-level, allowing to leverage the diversity in explanations to find "shortcuts" through the learning paths.

Figure 10.11 provides additional insightful examples of the generated sequences extracted from the term-coverage data of each sequence. Figure 10.11(d) is another example where the target document is a fairly advanced introduction

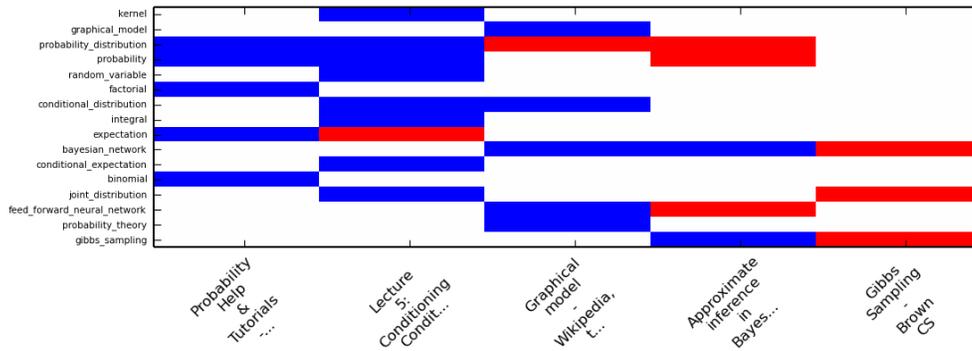


Figure 10.13: An example sequence of documents (columns) that lead the user to the goal document on the topic of *Gibbs Sampling*

to the topic (*Expectation Maximization*), which is preceded by a more gentle introduction to the same topic, as well as an additional prerequisite (*Maximum Likelihood*) which is a common prerequisite for this topic. Note, however, that while *Maximum Likelihood* is traditionally considered as a prerequisite for learning about *Expectation Maximization*, it is not the case for the more basic introduction to this topic (*What is the Expectation Maximization algorithm*), as that particular introduction aims to bring a very high-level understanding of the topic without burdening the reader with additional prerequisite requirements. Therefore, in that particular sequence, the reader is first given a gentle introduction to the topic, then the necessary prerequisite (*Maximum Likelihood*) for understanding the more advanced introduction.

### Error analysis

The extracted sequences are not without errors. These errors stem from several potential sources, as a fairly involved pipeline lies between the raw document and the resulting optimal sequence, providing an opportunity for errors to propagate

through the different stages. We break down these errors by their source to give a better understanding of how these problems need to be addressed in future work:

**Terminology extraction:** The greatest source of errors stems from errors in terminology extraction. There are two types of errors involved in terminology extraction: *false negatives* (missing terms) and *false positives* (term sense disambiguation errors). False negatives are more difficult to detect and often result in missing prerequisites; missing terms are especially difficult when relying on a finite vocabulary list. Addressing this issue generally requires a general terminology extraction method that can adapt to out-of-domain terminology. The issue with this solution, however, is that a more general method will result in more false positives, which leads to a whole lot of other issues. False positives occur when either a common English word is mistaken for a term or when two terms from different domains or with different meaning are mistaken for one another. This leads to downstream problems like the example in Figure 10.11e. In this example, the term *neural network* is not used in a conventional sense, and primarily refers to a broad class of methods that do not require the prerequisite knowledge of *neural networks*. This term propagates to the optimization, resulting in an unnecessary prerequisite reading about the *back-propagation algorithm*. These errors are especially prevalent with abbreviations, such as *bic* as *em*. Overcoming this challenge requires the model's awareness of word-senses, disambiguating which would rely on the context of these terms. Word-sense disambiguation is an active area of research in the field of Natural Language Processing, which can likely be leveraged to address this issue.

**Explain/Assume classification:** The second greatest source of errors are the mis-

takes made by the aspect classifier. Classifying an explained term as an assumed term creates unnecessary prerequisites, while the reverse results in missing potentially important prerequisites. The example in Figure 10.13 serves as an illustration. Although the original target document labels the concept of *gibbs sampling* as assumed, the content of that web-page is actually explaining this concept, while in the second to last document in the sequence where the concept is classified as *explained*, *gibbs sampling* is actually assumed to be known. The reason for the error is partly because both web-pages are fairly short, and the classifier's estimates of the aspects have a fairly large variance. When such errors are made in only a few terms in an otherwise sufficiently large document, the goal is that the resulting optimization's sensitivity to these errors is reduced. This brings us to the importance of robustness in the optimization problem, discussed next.

**Path optimization:** because we solve the optimization problem exactly (i.e., find a global optimum), there are no errors stemming from the optimization itself (this will become a potential source of errors, however, when an approximation scheme, e.g., LP rounding, is used to obtain an approximate solution). However, the formulation of the optimization problem can be improved so as to introduce robustness to the errors in the earlier stages of the pipeline. As path optimization is the final stage that produces the final output, its sensitivity to the errors in terminology extraction and term aspect classification are directly reflected in the resulting output. Introducing robustness to these errors directly in the formulation of the optimization problem is potentially the most effective way to address the issues in the earlier stages of the pipeline. One issue with the current formulation is its inability to incorporate the relative frequency of the term into the optimization objective: ideally terms that appear less frequently in

a document should have a lesser precedence for coverage than those that appear more frequently (Assumption 2 in the Introduction). The example in Figure 10.10 demonstrates the lack of robustness in the third document (D), where the appearance of the term *integral* creates an additional sequence of documents that cover this concept. From our earlier analysis in Section 10.3, we have shown that the errors in the *Explain/Assume* classifier are directly related to the relative frequency of the terms, and thus a way to incorporate these frequencies as weights into the optimization would potentially be the most effective way to deal with this noise.

## 10.9 Conclusion

We developed what we believe is the first end-to-end approach towards automatic *curriculum extraction* from the web, relying on the following pipeline: (i) extracting what is assumed vs. what is explained in a single document and then (ii) connecting these documents into a sequence ensuring that the progression builds up the knowledge of the learner gradually towards their goal. We developed algorithms that addressed both of these components: (i) a semi-supervised approach for learning a term aspect classifier from a very small set of annotated examples and (ii) an optimization problem for learning path recommendation based on the user's learning goals. To the best of our knowledge, we for the first time demonstrate and leverage the most unique characteristic of the web in the domain of learning: *diversity*, i.e., presence of content that explains the same concepts but in many different ways and from many different angles. This property of the web opens the doors to personalizing learning sequences that leverage the differences in explanations to find the most effective paths and shortcuts

through the Internet. Finally, we outlined a set of important challenges that need to be addressed in order to make this task a practical reality at web-scale. We hope that the work described in this chapter, in addition to the datasets that we release, will serve to inspire interest from the community in what we believe is a challenging and an important task.

## CHAPTER 11

### CONCLUSION AND FUTURE WORK

In this dissertation, we have addressed and pursued a number of opportunities and challenges that arise as a consequence of scaling in education. Building on an extensive body of work in automating instruction (machine teaching), one of the distinguishing themes that unites the contributions of this thesis is the notion of a *self-sustainable classroom* — that is leveraging the crowd of learners in order to generate and utilize its own learning content, with minimal instructor input. A self-sustainable classroom relies on *democratizing* the traditional dichotomy of roles of learners and instructors — relaxing the boundary that separates them in order to allow everyone to contribute content (such as assessments and learning content). The motivation for allowing everyone to contribute stems from the recent growth in our classrooms — an effect largely contributed to by the virtualization of classrooms via platforms such as Massive open online courses (MOOCs) and public learning platforms like Khan Academy. A distinguishing feature of these web-scale classrooms is a heavily disproportionate student to teacher ratio. By allowing and encouraging learners to become active contributors to the content, we address two key challenges that arise in large and growing classrooms:

1. **scalability**: leveraging learners as instructors multiplies the scaling capacity of the classroom in tasks such as grading and providing feedback.
2. **diversity**: as classrooms grow in size and geographic diversity, so does the diversity of the learners in their skills, backgrounds and learning styles. Catering to diverse learners requires diverse content that can match the backgrounds and needs of these learners.

By delegating the capacity of an instructor to every student (e.g. in allowing students to contribute original learning content, generate questions and grade submissions of other students) we introduce the challenge of automatically and intelligently “filtering” the students’ noisy input in a way that leverages the input from able students and discounts input from less able students. Moreover, as classrooms grow in size, relying on instructors becomes ever more costly and difficult, favoring methods that are able to automatically identify students’ abilities and exploit that information to effectively filter content. In this thesis, we focused on the development of such methods.

## **11.1 Summary of contributions**

More specifically, our contribution is in developing synergistic methods that combine automation and crowdsourcing to scale the process of *mastery learning* — a feedback loop of continuous assessment and instruction that ensures a stable progression of the student’s learning. The two components of mastery learning are *assessment* and *learning content curation*. In this section, we briefly summarize our contributions to each component.

### **11.1.1 Assessment**

#### **Question generation from text**

We proposed and evaluated a method for automatic question generation from free text – providing a scalable approach to generating assessment content that

learners and instructors can use to assess comprehension.

### **Multiple choice question generation**

We proposed and evaluated a method for generating multiple choice questions that adapt their choice sets to a given population of learners in order to maximize information about the learners' ability.

### **Joint assessment and grading**

We proposed and evaluated a method for grading open-ended submissions by (i) aggregating them into multiple choice questions and (ii) using the responses to these multiple choice questions by other students to automatically grade and assess all students. Our method presents as an alternative to peer-grading that offers additional benefits: (i) by re-framing the task of grading as additional testing, students are incentivized to put effort into grading implicitly (by being motivated to answer questions correctly) and (ii) answers submitted by students to the multiple choice questions can be used to cluster similar open-response submissions (see below).

### **Solution clustering via implicit feedback**

We proposed and evaluated a method that leverages the answers submitted to multiple choice questions with options constructed from open-response submissions to (i) automatically identify clusters of semantically similar submissions and (ii) grade them, generating a (i) potentially useful signal to instructors about

common misconceptions in the classroom and (ii) amplifying grader effort.

### **Self-grading**

We proposed and evaluated a method for scalable and calibrated self-assessment – an approach to grading that relies on students assigning grades to themselves (as an alternative to peer-grading). We utilize a mechanism that encourages honest self-grading and at the same time facilitates estimation of learners’ mastery which can be calibrated with a limited amount of instructor input.

## **11.1.2 Learning content curation**

### **Language learning from the web**

We proposed and evaluated a method for leveraging web content (e.g. Wikipedia articles) to implicitly teach readers vocabulary in a foreign language by exposing them to foreign words in the context of a text written in their native language. We developed a model and an optimization problem for automatically selecting articles and translating subsets of their words with an explicit objective of maximizing the reader’s acquisition of new words.

### **Learning content representation and personalized lesson sequencing**

We proposed and evaluated a method for leveraging logs of students’ interactions with learning content (e.g. lessons and assessment modules) to induce a joint representation (embedding) of students and learning content. We demonstrate

that the induced representation is capable of predicting effective paths through learning content.

### **Curating learning paths through heterogeneous web resources**

We proposed and evaluated methods for (i) automatically identifying the explained and assumed concepts in the heterogeneous learning content on the web (e.g. tutorials, lecture notes, blogs, book chapters) and (ii) automatically joining this content into learning paths that connect web resources that explain concepts to the resources where those concepts are assumed.

## **11.2 Future directions**

In this section, we conclude with a speculation of what the field of *machine teaching* may look like in the next decade. As we have done consistently through this dissertation, we organize it according to the two components of the mastery learning paradigm: *assessment* and *learning content curation*.

### **11.2.1 Growth of implicit assessment**

With the proliferation of user-contributed content, the logs of user interactions with that content carry an intrinsic value that so far had received little attention from the perspective of assessment. Implicit signal from learners' interactions with the content and other learners carries the potential to reveal their underlying strengths and weaknesses, areas where they require remediation and the kind

of remediation that they may require. This implicit signal may be in the context of technical and classroom forums such as StackExchange, Quora and Piazza, where the implicit signal includes “who answered whose question”, “how many upvotes the answer received and from what other learners”, “what answers the learner upvoted as correct”. The value of implicit signal such as click-through had long ago been demonstrated to be valuable in the context of preference elicitation on the web [84] and had tremendous impact in information retrieval. We believe that implicit interactions in the context of learning have the potential to bring a similar level of impact in learning analytics. Implicit signal carries the advantage of being continuous and non-intrusive, in contrast to explicit tests which are sparse, high stakes and as a result may be noisy in their assessment. The value of implicit signal became evident in the context of information retrieval with the proliferation of search engines and interaction logs that made implicit signal abundant. With the scaling of learning to the web, similar growth in interactions will likely open important research questions such as: (i) how to develop mechanisms that encourage truthful behavior and elicit maximum information about learners’ proficiencies and deficiencies and (ii) how to combine limited explicit and implicit assessment to obtain reliable, calibrated assessment of students’ learning. We believe that questions such as these provide a rich and fruitful direction for future research in assessment.

## **11.2.2 Social organization of crowdsourcing efforts for learning content on the web**

The web is the only collection of resources today that can cater to the growing diversity of web-scale classrooms. The reason for this is that the web, unlike a textbook or a course, contains an extensive amount of diversity in its content, i.e. content that explains the same concepts but in many different ways and from different perspectives. We believe that leveraging the constantly up-to-date, growing pool of learning content on the web is the next frontier in the evolution of *machine teaching*. As research into leveraging the web for curating adaptable learning content continues, and as platforms for organizing the web for the purpose of teaching and learning become available, it is possible that the web itself will adapt to this new paradigm of web-scale instruction. We briefly share our speculation of what the “learning web” may look like, and the research problems that will arise.

### **User-generated content**

In the long term, we believe that the services that help learners navigate the existing tutorial content on the web, will facilitate a natural outgrowth of an ecosystem of user-generated tutorial content that does not exist anywhere on the web today. The reason for this is the potential of such platforms to generate a social incentive system and a centralized hub around user-generated tutorial content, as a result of which more people will be inspired to contribute. This is not unlike Wikipedia, StackExchange and Quora, whose main assets are their content, and social incentive plays a central role in incentivizing quality and

volume of contribution.

Platforms like this will provide additional exposure to learning content, as it will be catered to the learners for whom this content is most appropriate. Content creators will have the ability to augment their content with richer structured information, in order to increase its visibility for the right learners at the right time, e.g. by directly specifying a set of prerequisites concept tags.

Tools that help users navigate that content and establish explicit and implicit reward mechanisms, i.e. via explicit feedback that learners provide towards the content that they navigate, will create a competitive setting that encourages people to contribute more content. What will fundamentally distinguish platforms of this sort from Wikipedia and the alike, will be the inherent diversity and redundancy of explanations, e.g. hundreds of articles about the concept of “neural network”, each catering to a different set of learners. We believe that a platform that facilitates learning by organizing the web of diverse educational resources and aiding users’ navigation through this content will facilitate the growth of these resources as a side-effect, potentially creating a self-reinforcing ecosystem of users and content generators.

### **Rich feedback towards existing content**

We believe that the value of the learning resources on the web stems as much from their actual content as from our understanding of what that content is about (i.e. having some degree of understanding of what is explained and assumed in a given page). Similar to how a traditional search engine adds value to a web-page by being able to retrieve it in response to the appropriate queries, a

platform that is aware of which web resources explain which concepts and with what quality, is potentially as valuable as the content itself. As users navigate the web resources on this platform, they will leave a rich trace of both implicit and explicit feedback about the material and themselves. Implicit feedback consists of features such as whether a particular query or page was abandoned, or if a particular recommended path was followed. Explicit feedback consists of explicit evaluation (e.g. "like/dislike") of a recommended page or a sequence that the user provides to the system. This feedback provides an important assessment of the material that places it in the context of all other content, allowing for appropriate and personalized recommendation. As platforms like this grow in their adoption, so will the logs of such implicit interactions, creating a rich direction for new streams of research.

## BIBLIOGRAPHY

- [1] Heike Adel, Ngoc Thang Vu, Franziska Kraus, Tim Schlippe, Haizhou Li, and Tanja Schultz. Recurrent neural network language modeling for code switching conversational speech. *ICASSP*, 2012.
- [2] Manish Agarwal, Rakshit Shah, and Prashanth Mannem. Automatic question generation using discourse cues. In Proceedings of the 6th Workshop on Innovative Use of NLP for Building Educational Applications, pages 1–9. Association for Computational Linguistics, 2011.
- [3] Rakesh Agrawal, Sunandan Chakraborty, Sreenivas Gollapudi, Anitha Kannan, and Krishnaram Kenthapadi. Empowering authors to diagnose comprehension burden in textbooks. In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 967–975. ACM, 2012.
- [4] Rakesh Agrawal, Sreenivas Gollapudi, Anitha Kannan, and Krishnaram Kenthapadi. Study navigator: An algorithmically generated aid for learning from electronic textbooks. *JEDM-Journal of Educational Data Mining*, 6(1):53–75, 2014.
- [5] Richard C Anderson and W Barry Biddle. On asking people questions about what they are reading. *Psychology of learning and motivation*, 9:89–132, 1975.
- [6] Thomas Andre. Does answering higher-level questions while reading facilitate productive learning? *Review of Educational Research*, 49(2):280–318, 1979.
- [7] Yoram Bachrach, Thore Graepel, Tom Minka, and John Guiver. How to grade a test without knowing the answers—a bayesian graphical model for adaptive crowdsourcing and aptitude testing. arXiv preprint arXiv:1206.6386, 2012.
- [8] Richard Baraniuk. Openstax.
- [9] Sumit Basu, Chuck Jacobs, and Lucy Vanderwende. Powergrading: a clustering approach to amplify human effort for short answer grading. *Transactions of the Association for Computational Linguistics*, 1:391–402, 2013.

- [10] Sumit Basu, Chuck Jacobs, and Lucy Vanderwende. Powergrading: a clustering approach to amplify human effort for short answer grading. Transactions of the Association for Computational Linguistics, 1:391–402, 2013.
- [11] Lee Becker, Sumit Basu, and Lucy Vanderwende. Mind the gap: learning to choose gaps for question generation. In Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 742–751. Association for Computational Linguistics, 2012.
- [12] Isaac Bejar. A sentence-based automated approach to the assessment of writing: A feasibility study. Machine-Mediated Learning, 2(4):321–332, 1987.
- [13] Hedi M Belazi, Edward J Rubin, and Almeida Jacqueline Toribio. Code switching and x-bar theory: The functional head constraint. Linguistic inquiry, pages 221–237, 1994.
- [14] Austin R Benson, Ravi Kumar, and Andrew Tomkins. On the relevance of irrelevant alternatives. In Proceedings of the 25th International Conference on World Wide Web, pages 963–973. International World Wide Web Conferences Steering Committee, 2016.
- [15] Rakesh Mohan Bhatt. Code-switching, constraints, and optimal grammars. Lingua, 102(4):223–251, 1997.
- [16] J Eric Bickel. Scoring rules and decision analysis education. Decision Analysis, 7(4):346–357, 2010.
- [17] Or Biran, Samuel Brody, and Noemie Elhadad. Putting it simply: a context-aware approach to lexical simplification. 2011.
- [18] Fabian Blacher. SMT-based Text Generation for Code-Switching Language Models. PhD thesis, Nanyang Technological University, Singapore, 2011.
- [19] David M Blei and Peter I Frazier. Distance dependent Chinese restaurant processes. The Journal of Machine Learning Research, 12:2461–2488, 2011.
- [20] Benjamin S Bloom. Human characteristics and school learning. McGraw-Hill, 1976.

- [21] Arielle Borovsky, Jeffrey L Elman, and Marta Kutas. Once is enough: N400 indexes semantic integration of novel word meanings from a single exposure in context. Language Learning and Development, 8(3):278–302, 2012.
- [22] Timothy F Bresnahan, Scott Stern, and Manuel Trajtenberg. Market segmentation and the sources of rents from innovation: Personal computers in the late 1980’s. Technical report, National bureau of economic research, 1996.
- [23] Glenn W Brier. Verification of forecasts expressed in terms of probability. Monthly weather review, 78(1):1–3, 1950.
- [24] Michael Brooks, Sumit Basu, Charles Jacobs, and Lucy Vanderwende. Divide and correct: using clusters to grade short answers at scale. In Proceedings of the first ACM conference on Learning@ scale conference, pages 89–98. ACM, 2014.
- [25] Michael Brooks, Sumit Basu, Charles Jacobs, and Lucy Vanderwende. Divide and correct: using clusters to grade short answers at scale. In Proceedings of the first ACM conference on Learning@ scale conference, pages 89–98. ACM, 2014.
- [26] Jonathan C Brown, Gwen A Frishkoff, and Maxine Eskenazi. Automatic question generation for vocabulary assessment. In Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, pages 819–826. Association for Computational Linguistics, 2005.
- [27] Peter Brusilovsky and David W Cooper. Domain, task, and user models for an adaptive hypermedia performance support system. In Proceedings of the 7th international conference on Intelligent user interfaces, pages 23–30. ACM, 2002.
- [28] Peter Brusilovsky, John Eklund, and Elmar Schwarz. Web-based education for all: a tool for development adaptive courseware. Computer networks and ISDN systems, 30(1):291–300, 1998.
- [29] Peter Brusilovsky and Nicola Henze. Open corpus adaptive educational hypermedia. In The adaptive web, pages 671–696. Springer, 2007.
- [30] Leonie Burgess and Deborah J Street. Optimal designs for 2 k choice experiments. 2003.

- [31] Marco Caliendo and Sabine Kopeinig. Some practical guidance for the implementation of propensity score matching. Journal of economic surveys, 22(1):31–72, 2008.
- [32] John B Carroll. A model of school learning. Teachers college record, 1963.
- [33] Shuo Chen, Joshua L Moore, Douglas Turnbull, and Thorsten Joachims. Playlist prediction via metric embedding. In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 714–722. ACM, 2012.
- [34] W Chen. Aist, g., mostow, j.: Generating questions automatically from informational text. In Proceedings of the 2nd Workshop on Question Generation, held at the Conference on AI in, Education, pages 17–24, 2009.
- [35] Chaushie Chu. A paired combinatorial logit model for travel demand analysis. In Transport Policy, Management & Technology Towards 2001: Selected Proceedings of the Fifth World Conference on Transport Research, volume 4, 1989.
- [36] Tom Cobb. Computing the vocabulary demands of l2 reading. Language Learning & Technology, 11(3):38–63, 2007.
- [37] R Dennis Cook and Christopher J Nachtrheim. A comparison of algorithms for constructing exact d-optimal designs. Technometrics, 22(3):315–324, 1980.
- [38] Clyde H Coombs. Psychological scaling without a unit of measurement. Psychological review, 57(3):145, 1950.
- [39] Albert T Corbett and John R Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. User modeling and user-adapted interaction, 4(4):253–278, 1994.
- [40] Albert T. Corbett and John R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. User Modeling and User-Adapted Interaction, 4(4):253–278, 1994.
- [41] Sérgio Curto, Ana Cristina Mendes, and Luísa Coheur. Exploring linguistically-rich patterns for question generation. In Proceedings of the UCNLG+ eval: Language Generation and Evaluation Workshop, pages 33–38. Association for Computational Linguistics, 2011.

- [42] Alexander Philip Dawid and Allan M Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. Applied statistics, pages 20–28, 1979.
- [43] Marie Jean Antoine Nicolas de Caritat et al. Essai sur l’application de l’analyse à la probabilité des décisions rendues à la pluralité des voix. L’imprimerie royale, 1785.
- [44] Michel C Desmarais and Ryan SJ d Baker. A review of recent advances in learner and skill modeling in intelligent learning environments. User Modeling and User-Adapted Interaction, 22(1-2):9–38, 2012.
- [45] Matthew T Downey. Ben D. Wood: Educational Reformer. Educational Testing Service, 1965.
- [46] Martin Ebner, Elke Lackner, and Michael Kopp. How to mooc?-a pedagogical guideline for practitioners. In The International Scientific Conference eLearning and Software for Education, volume 4, page 215. " Carol I" National Defence University, 2014.
- [47] Chaitanya Ekanadham and Yan Karklin. T-skirt: Online estimation of student proficiency in an adaptive learning system. Machine Learning for Education Workshop at ICML, 2015.
- [48] Noemie Elhadad and Komal Sutaria. Mining a lexicon of technical terms and lay equivalents. In Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing, pages 49–56. Association for Computational Linguistics, 2007.
- [49] Nick C Ellis. Consciousness in second language acquisition: A review of field studies and laboratory experiments. Language awareness, 4(3):123–146, 1995.
- [50] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. JLMR, 9:1871–1874, 2008.
- [51] Valerii Vadimovich Fedorov. Theory of optimal experiments. Elsevier, 1972.
- [52] Jose Ferreira. Knewton. <http://http://www.knewton.org>.

- [53] Gerhard H Fischer and Ivo W Molenaar. Rasch models: Foundations, recent developments, and applications. Springer Science & Business Media, 2012.
- [54] Ronald A Fisher. The arrangement of field experiments. In Breakthroughs in statistics, pages 82–91. Springer, 1992.
- [55] Sir Ronald Aylmer Fisher, Ronald Aylmer Fisher, Statistiker Genetiker, Ronald Aylmer Fisher, Statistician Genetician, Great Britain, Ronald Aylmer Fisher, and Statisticien Généticien. The design of experiments. 1960.
- [56] R. Fortet. L’Algèbre de Boole et ses applications en recherche opérationnelle. Cahiers Centre Etudes Rech. Oper. no., 4:5–36, 1959.
- [57] Stefan L Frank, Leun J Otten, Giulia Galli, and Gabriella Vigliocco. Word surprisal predicts n400 amplitude during reading. In Proceedings of the 51st annual meeting of the Association for Computational Linguistics, pages 878–883, 2013.
- [58] Francis Galton. Vox populi (the wisdom of crowds). Nature, 75(7):450–451, 1907.
- [59] Fred Genesee. Bilingual first language acquisition: Exploring the limits of the language faculty. Annual Review of Applied Linguistics, 21:153–168, 2001.
- [60] Elena L Glassman, Rishabh Singh, and Robert C Miller. Feature engineering for clustering student solutions. In Proceedings of the first ACM conference on Learning@ scale conference, pages 171–172. ACM, 2014.
- [61] Mark E Glickman. Parameter estimation in large dynamic paired comparison experiments. Applied Statistics, pages 377–394, 1999.
- [62] Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. Journal of the American Statistical Association, 102(477):359–378, 2007.
- [63] Nadav Golbandi, Yehuda Koren, and Ronny Lempel. Adaptive bootstrapping of recommender systems using decision trees. In Proceedings of the fourth ACM international conference on Web search and data mining, pages 595–604. ACM, 2011.

- [64] Wael H Gomaa and Aly A Fahmy. Short answer grading using string similarity and corpus-based similarity. International Journal of Advanced Computer Science and Applications (IJACSA), 3(11), 2012.
- [65] JP González-Brenes, Yun Huang, and Peter Brusilovsky. General features in knowledge tracing: Applications to multiple subskills, temporal item response theory, and expert knowledge. In Proceedings of the 7th International Conference on Educational Data Mining (accepted, 2014), 2014.
- [66] Roger Grosse. Metacademy. <http://www.metacademy.org>.
- [67] Thomas M Haladyna. Writing Test Items To Evaluate Higher Order Thinking. ERIC, 1997.
- [68] Thomas M Haladyna and Steven M Downing. Validity of a taxonomy of multiple-choice item-writing rules. Applied Measurement in Education, 2(1):51–78, 1989.
- [69] Thomas M Haladyna and Steven M Downing. How many options is enough for a multiple-choice test item? Educational and Psychological Measurement, 53(4):999–1010, 1993.
- [70] Thomas M Haladyna, Steven M Downing, and Michael C Rodriguez. A review of multiple-choice item-writing guidelines for classroom assessment. Applied measurement in education, 15(3):309–333, 2002.
- [71] Peter L. Hammer and Sergiu Rudeanu. Boolean methods in operations research and related areas. Springer-Verlag New York, Inc., New York, 1968.
- [72] Glenn W Harrison, Jimmy Martínez-Correa, and J Todd Swarthout. Inducing risk neutral preferences with binary lotteries: A reconsideration. Journal of Economic Behavior & Organization, 94:145–159, 2013.
- [73] Margot Haynes and Ilu Baker. American and chinese readers learning from lexical familiarization in english texts. Second language reading and vocabulary learning, pages 130–152, 1993.
- [74] Margot Haynes and Ilu Baker. American and chinese readers learning from lexical familiarization in english texts. Second language reading and vocabulary learning, pages 130–152, 1993.

- [75] Michael Heilman and Noah A Smith. Good question! statistical ranking for question generation. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pages 609–617. Association for Computational Linguistics, 2010.
- [76] Ralf Herbrich, Tom Minka, and Thore Graepel. Trueskill: A bayesian skill rating system. In Advances in neural information processing systems, pages 569–576, 2006.
- [77] David Hirsh, Paul Nation, et al. What vocabulary size is needed to read unsimplified texts for pleasure? Reading in a foreign language, 8:689–689, 1992.
- [78] Marlise Horst, Tom Cobb, Thomas Cobb, and Paul Meara. Beyond a clock-work orange: Acquiring second language vocabulary through reading. Reading in a foreign language, 11(2):207–223, 1998.
- [79] ME Horst. Text encounters of the frequent kind: Learning L2 vocabulary through reading. PhD thesis, University of Wales Swansea, 2001.
- [80] Joel Huber and Klaus Zwerina. The importance of utility balance in efficient choice designs. Journal of Marketing research, pages 307–317, 1996.
- [81] Thomas Huckin and James Coady. Incidental vocabulary acquisition in a second language. Studies in second language acquisition, 21(02):181–193, 1999.
- [82] Nguyen Quoc Viet Hung, Nguyen Thanh Tam, Lam Ngoc Tran, and Karl Aberer. An evaluation of aggregation techniques in crowdsourcing. In Web Information Systems Engineering–WISE 2013, pages 1–15. Springer, 2013.
- [83] Ben Hutchinson. Modelling the substitutability of discourse connectives. In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, pages 149–156. Association for Computational Linguistics, 2005.
- [84] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. Accurately interpreting clickthrough data as implicit feedback. In Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, pages 154–161. Acm, 2005.

- [85] Thorsten Joachims and Karthik Raman. Bayesian ordinal aggregation of peer assessments: A case study on kdd 2015. Technical report, Technical report, Cornell University, 2015.
- [86] Sally Jordan and Tom Mitchell. e-Assessment for learning? The potential of short-answer free-text questions with tailored feedback. British Journal of Educational Technology, 40(2):371–385, 2009.
- [87] Robert M Kaplan and Dennis P Saccuzzo. Psychological testing: Principles, applications, and issues. Cengage Learning, 2012.
- [88] David Kauchak. Improving text simplification language modeling using unsimplified text data. In Proceedings of ACL, 2013.
- [89] Roselinde Kessels, Bradley Jones, Peter Goos, and Martina Vandebroek. An efficient algorithm for constructing bayesian optimal choice designs. Journal of Business & Economic Statistics, 27(2):279–291, 2009.
- [90] Juho Kim et al. Learnersourcing: improving learning with collective learner activity. PhD thesis, Massachusetts Institute of Technology, 2015.
- [91] Knewton. The knewton platform: A general-purpose adaptive learning infrastructure. Technical report, 2015.
- [92] Kenneth R Koedinger, John R Anderson, William H Hadley, and Mary A Mark. Intelligent tutoring goes to school in the big city. 1997.
- [93] Frank Koppelman and Chieh-Hua Wen. Nested logit models which are you using? Transportation Research Record: Journal of the Transportation Research Board, (1645):1–7, 1998.
- [94] Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. Journal of Machine Learning Research, 9(Feb):235–284, 2008.
- [95] Chinmay Kulkarni, Koh Pang Wei, Huy Le, Daniel Chia, Kathryn Papadopoulos, Justin Cheng, Daphne Koller, and Scott R Klemmer. Peer and self assessment in massive online classes. In Design Thinking Research, pages 131–168. Springer, 2015.

- [96] Marta Kutas and Steven A Hillyard. Brain potentials during reading reflect word expectancy and semantic association. Nature, 1984.
- [97] Igor Labutov, Sumit Basu, and Lucy Vanderwende. Deep questions without deep understanding. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, volume 1, pages 889–898.
- [98] Igor Labutov and Hod Lipson. Web as a textbook: Curating targeted learning paths through the heterogeneous learning resources on the web. 2016.
- [99] Igor Labutov, Kelvin Luu, Hod Lipson, and Christoph Studer. Optimally discriminative choice sets in discrete choice models: Application to data-driven test design. In Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2016.
- [100] Igor Labutov and Christoph Studer. Calibrated self-assessment. 2016.
- [101] Andrew S Lan, Christoph Studer, and Richard G Baraniuk. Time-varying learning and content analytics via sparse factor analysis. In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 452–461. ACM, 2014.
- [102] Andrew S Lan, Divyanshu Vats, Andrew E Waters, and Richard G Baraniuk. Mathematical language processing: Automatic grading and feedback for open response mathematical questions. In Proceedings of the Second (2015) ACM Conference on Learning@ Scale, pages 167–176. ACM, 2015.
- [103] Andrew S Lan, Divyanshu Vats, Andrew E Waters, and Richard G Baraniuk. Mathematical language processing: Automatic grading and feedback for open response mathematical questions. In Proceedings of the Second (2015) ACM Conference on Learning@ Scale, pages 167–176. ACM, 2015.
- [104] Andrew S Lan, Andrew E Waters, Christoph Studer, and Richard G Baraniuk. Sparse factor analysis for learning and content analytics. The Journal of Machine Learning Research, 15(1):1959–2008, 2014.
- [105] Batia Laufer. The lexical plight in second language reading: Words you dont know, words you think you know, and words you cant guess. Second language vocabulary acquisition, 3034, 1997.

- [106] Claudia Leacock and Martin Chodorow. C-rater: Automated scoring of short-answer questions. Computers and the Humanities, 37(4):389–405, 2003.
- [107] Chen Liang, Zhaohui Wu, Wenyi Huang, and C Lee Giles. Measuring prerequisite relations among concepts.
- [108] Yi-ling Lin and Peter Brusilovsky. Towards open corpus adaptive hypermedia: a study of novelty detection approaches. In International Conference on User Modeling, Adaptation, and Personalization, pages 353–358. Springer, 2011.
- [109] David Lindberg, Fred Popowich, John Nesbit, and Phil Winne. Generating natural language questions to support learning on-line. 2013.
- [110] WJ Linden and Ronald K Hambleton. Handbook of modern item response theory. New York, 1997.
- [111] John M Lipski. Code-switching or borrowing? no sé so no puedo decir, you know. In Selected Proceedings of the Second Workshop on Spanish Sociolinguistics, pages 1–15, 2005.
- [112] Frederic M Lord. Applications of item response theory to practical testing problems. Routledge, 1980.
- [113] Frederic M Lord, Melvin R Novick, and Allan Birnbaum. Statistical theories of mental test scores. 1968.
- [114] R Duncan Luce. On the possible psychophysical laws. Psychological review, 66(2):81, 1959.
- [115] AA Lumsdaine. Teaching machines and self-instructional materials. Educational Technology Research and Development, 7(3):163–181, 1959.
- [116] Ernesto Macaro. Codeswitching in the l2 classroom: A communication and learning strategy. In Non-native language teachers, pages 63–84. Springer, 2005.
- [117] Prashanth Mannem, Rashmi Prasad, and Aravind Joshi. Question generation from paragraphs at upenn: Qgstec system description. In Proceedings of QG2010: The Third Workshop on Question Generation, pages 84–91, 2010.

- [118] Karen Mazidi and Rodney D Nielsen. Linguistic considerations in automatic question generation. In ACL (2), pages 321–326, 2014.
- [119] GI McCalla, DR Peachey, and B Ward. An architecture for the design of large-scale intelligent teaching systems. In Proceedings of the 4th National Conference of the CSCSI, pages 85–91, 1982.
- [120] Daniel McFadden. Modeling the choice of residential location. Transportation Research Record, (673), 1978.
- [121] Daniel McFadden et al. Conditional logit analysis of qualitative choice behavior. 1973.
- [122] James H McMillan. Secondary teachers' classroom assessment and grading practices. Educational Measurement: Issues and Practice, 20(1):20–32, 2001.
- [123] Sara McNeil. A hypertext history of instructional design. <http://faculty.coe.uh.edu/smcneil/cuin6373/idhistory/index.html>. (Accessed on 06/13/2016).
- [124] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems, pages 3111–3119, 2013.
- [125] Alan J Miller and Nam-Ky Nguyen. Algorithm as 295: A fedorov exchange algorithm for d-optimal design. Journal of the royal statistical society. series c (applied statistics), 43(4):669–677, 1994.
- [126] Lesley Milroy and Pieter Muysken. One speaker, two languages: Cross-disciplinary perspectives on code-switching. Cambridge University Press, 1995.
- [127] Tom Minka, John Winn, John Guiver, and David Knowles. Infer .net 2.5. Microsoft Research Cambridge, 2012.
- [128] Ruslan Mitkov, Le An Ha, and Nikiforos Karamanis. A computer-aided environment for generating multiple-choice test items. Natural Language Engineering, 12(02):177–194, 2006.

- [129] Ruslan Mitkov and Le An Ha. Computer-aided generation of multiple-choice tests. In Proceedings of the HLT-NAACL 03 workshop on Building educational applications using natural language processing-Volume 2, pages 17–22. Association for Computational Linguistics, 2003.
- [130] Michael Mohler and Rada Mihalcea. Text-to-text semantic similarity for automatic short answer grading. In Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics, pages 567–575. Association for Computational Linguistics, 2009.
- [131] Jan-Arjen Mondria, Marijke Wit-de Boer, et al. The effects of contextual richness on the guessability and the retention of words in a foreign language1. Applied linguistics, 12(3):249–267, 1991.
- [132] Visvaganthie Moodley. Code-switching and communicative competence in the language classroom. Journal for Language Teaching, 44(1):7–22, 2010.
- [133] Allan H Murphy and Robert L Winkler. Scoring rules in probability assessment and evaluation. Acta psychologica, 34:273–286, 1970.
- [134] Radford M Neal. Markov chain sampling methods for Dirichlet process mixture models. Journal of computational and graphical statistics, 9(2):249–265, 2000.
- [135] Andrew M Olney, Arthur C Graesser, and Natalie K Person. Question generation from concept maps. Dialogue and Discourse, 3(2):75–99, 2012.
- [136] Zachary Pardos, Yoav Bergner, Daniel Seaton, and David Pritchard. Adapting bayesian knowledge tracing to a massive open online course in edx. In Educational Data Mining 2013, 2013.
- [137] Zachary A Pardos and Neil T Heffernan. Tutor modeling vs. student modeling. In Proceedings of the Twenty-Fifth International Florida Artificial Intelligence Research Society Conference, pages 420–425, 2012.
- [138] Joonsuk Park and Kimberley Williams. The effects of peer-and self-assessment on the assessors. In Proceedings of the 47th ACM Technical Symposium on Computing Science Education, pages 249–254. ACM, 2016.
- [139] Kate Parry. Building a vocabulary through academic reading. Tesol Quarterly, pages 629–653, 1991.

- [140] Darwyn R Peachey and Gordon I McCalla. Using planning techniques in intelligent tutoring systems. International Journal of Man-Machine Studies, 24(1):77–98, 1986.
- [141] Charles S Peirce. Note on the theory of the economy of research (reprinted from 1876). Operations Research, 15(4):643–648, 1967.
- [142] Alexander Peysakhovich and Mikkel Plagborg-Møller. Proper scoring rules and risk aversion. Available at SSRN 2019078, 2012.
- [143] Chris Piech, Jonathan Huang, Zhenghao Chen, Chuong Do, Andrew Ng, and Daphne Koller. Tuned models of peer assessment in moocs. arXiv preprint arXiv:1307.2579, 2013.
- [144] Chris Piech, Jonathan Huang, Andy Nguyen, Mike Phulsuksombati, Mehran Sahami, and Leonidas Guibas. Learning program embeddings to propagate feedback on student code. arXiv preprint arXiv:1505.05969, 2015.
- [145] Dirk HJ Poot, J Arnold, Eric Achten, Marleen Verhoye, and Jan Sijbers. Optimal experimental design for diffusion kurtosis imaging. IEEE transactions on medical imaging, 29(3):819–829, 2010.
- [146] Lawrence Rabiner and B Juang. An introduction to hidden markov models. ieee assp magazine, 3(1):4–16, 1986.
- [147] Filip Radlinski and Thorsten Joachims. Active exploration for learning rankings from clickthrough data. In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 570–579. ACM, 2007.
- [148] Karthik Raman and Thorsten Joachims. Methods for ordinal peer grading. In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 1037–1046. ACM, 2014.
- [149] Georg Rasch. Probabilistic models for some intelligence and attainment tests. ERIC, 1993.
- [150] Georg Rasch. Probabilistic models for some intelligence and attainment tests. ERIC, 1993.
- [151] GJ Rath and Nancy S Anderson. The ibm research center teaching machine

- project:(1) the teaching of binary arithmetic; and (2) the simulation of a binary arithmetic teaching machine on the ibm 650. In US Air Force Office of Scientific Research Symposium on Teaching Machines, University of Pennsylvania, 1958.
- [152] GJ Rath and Nancy S Anderson. The ibm research center teaching machine project:(1) the teaching of binary arithmetic; and (2) the simulation of a binary arithmetic teaching machine on the ibm 650. In US Air Force Office of Scientific Research Symposium on Teaching Machines, University of Pennsylvania, 1958.
- [153] Paramesh Ray. Independence of irrelevant alternatives. Econometrica: Journal of the Econometric Society, pages 987–991, 1973.
- [154] M. D. Reckase. Multidimensional Item Response Theory. Springer Publishing Company, Incorporated, first edition, 2009.
- [155] Pierre Robillard. (0, 1) hyperbolic programming problems. Naval Research Logistics Quarterly, 18(1):47–57, 1971.
- [156] Peter Robinson. Attention, memory, and the noticing hypothesis. Language learning, 45(2):283–331, 1995.
- [157] Michael C Rodriguez. Three options are optimal for multiple-choice items: A meta-analysis of 80 years of research. Educational Measurement: Issues and Practice, 24(2):3–13, 2005.
- [158] Henry L Roediger III and Elizabeth J Marsh. The positive and negative consequences of multiple-choice testing. Journal of Experimental Psychology: Learning, Memory, and Cognition, 31(5):1155, 2005.
- [159] Cristian R Rojas, James S Welsh, Graham C Goodwin, and Arie Feuer. Robust optimal experiment design for system identification. Automatica, 43(6):993–1008, 2007.
- [160] Paul R Rosenbaum and Donald B Rubin. The central role of the propensity score in observational studies for causal effects. Biometrika, 70(1):41–55, 1983.
- [161] Vasile Rus, Brendan Wyse, Paul Piwek, Mihai Lintean, Svetlana Stoyanchev, and Cristian Moldovan. Overview of the first question generation

- shared task evaluation challenge. In Proceedings of the Third Workshop on Question Generation, pages 45–57, 2010.
- [162] Philip M Sadler and Eddie Good. The impact of self-and peer-grading on student learning. Educational assessment, 11(1):1–31, 2006.
- [163] Matthew J Salganik and Karen EC Levy. Wiki surveys: Open and quantifiable social data collection. PloS one, 10(5):e0123483, 2015.
- [164] Zsolt Sandor and Michel Wedel. Designing conjoint choice experiments using managers prior beliefs. Journal of Marketing Research, 38(4):430–444, 2001.
- [165] Leonard J Savage. Elicitation of personal probabilities and expectations. Journal of the American Statistical Association, 66(336):783–801, 1971.
- [166] Richard Schmidt. Awareness and second language acquisition. Annual review of applied linguistics, 13:206–226, 1992.
- [167] Richard C Schmidt and Richard W Schmidt. Attention and awareness in foreign language learning, volume 9. Natl Foreign Lg Resource Ctr, 1995.
- [168] Peter H Schönemann. On metric multidimensional unfolding. Psychometrika, 35(3):349–366, 1970.
- [169] Caroline Schouten-van Parreren. Vocabulary learning through reading: Which conditions should be met when presenting words in texts. AILA review, 6(1):75–85, 1989.
- [170] Hinrich Schütze. Introduction to information retrieval. In Proceedings of the international communication of association for computing machinery conference, 2008.
- [171] Lee Schwartz, Takako Aikawa, and Michel Pahud. Dynamic language learning tools. In InSTIL/ICALL Symposium 2004, 2004.
- [172] Reinhard Selten. Axiomatic characterization of the quadratic scoring rule. Experimental Economics, 1(1):43–62, 1998.
- [173] Reinhard Selten, Abdolkarim Sadrieh, and Klaus Abbink. Money does not induce risk neutral behavior, but binary lotteries do even worse. Theory and Decision, 46(3):213–252, 1999.

- [174] Nihar B Shah, Joseph Bradley, Sivaraman Balakrishnan, Abhay Parekh, Kannan Ramchandran, and Martin J Wainwright. Some scaling laws for MOOC assessments. In KDD Workshop on Data Mining for Educational Assessment and Feedback (ASSESS 2014), 2014.
- [175] Nihar B Shah, Joseph Bradley, Sivaraman Balakrishnan, Abhay Parekh, Kannan Ramchandran, and Martin J Wainwright. Some scaling laws for mooc assessments. In KDD Workshop on Data Mining for Educational Assessment and Feedback (ASSESS 2014), 2014.
- [176] Bruce Arne Sherwood. The tutor language. 1974.
- [177] Advait Siddharthan. Syntactic simplification and text cohesion. Research on Language and Computation, 4(1):77–109, 2006.
- [178] Advait Siddharthan, Ani Nenkova, and Kathleen McKeown. Syntactic simplification for improving content selection in multi-document summarization. In Proceedings of the 20th international conference on Computational Linguistics, page 896. Association for Computational Linguistics, 2004.
- [179] Burrhus Frederic Skinner. Teaching machines. The review of economics and statistics, pages 189–191, 1960.
- [180] Jascha Sohl-Dickstein. Temporal multi-dimensional item response theory, 2014.
- [181] Tamar Solorio and Yang Liu. Learning to predict code-switching points. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages 973–981. Association for Computational Linguistics, 2008.
- [182] Brent Strong, Mark Davis, and Val Hawks. Self-grading in large general education classes: A case study. College Teaching, 52(2):52–57, 2004.
- [183] Mingxuan Sun, Fuxin Li, Joonseok Lee, Ke Zhou, Guy Lebanon, and Hongyuan Zha. Learning multiple-question decision trees for cold-start recommendation. In Proceedings of the sixth ACM international conference on Web search and data mining, pages 445–454. ACM, 2013.
- [184] Partha Pratim Talukdar and William W Cohen. Crowdsourced comprehension: predicting prerequisite structure in wikipedia. In Proceedings of

the Seventh Workshop on Building Educational Applications Using NLP, pages 307–315. Association for Computational Linguistics, 2012.

- [185] Stacey Terman. GroverCode: Code Canonicalization and Clustering Applied to Grading. PhD thesis, Massachusetts Institute of Technology, 2016.
- [186] David Thissen, Lynne Steinberg, and Anne R Fitzpatrick. Multiple-choice models: The distractors are also part of the item. Journal of Educational Measurement, 26(2):161–176, 1989.
- [187] T Tian and M Yang. Efficiency of the coordinate-exchange algorithm in constructing exact optimal discrete choice experiments. Journal of Statistical Theory and Practice, (just-accepted), 2016.
- [188] Keith Topping. Self and peer assessment in school and university: Reliability, validity and utility. In Optimising new modes of assessment: In search of qualities and standards, pages 55–87. Springer, 2003.
- [189] Kenneth E Train. Discrete choice methods with simulation. Cambridge university press, 2009.
- [190] Andrew Trusty and Khai N Truong. Augmenting the web for second language vocabulary learning. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pages 3179–3188. ACM, 2011.
- [191] Harriet Tyson-Bernstein. A conspiracy of good intentions. america’s textbook fiasco. 1988.
- [192] Divyanshu Vats, Christoph Studer, Andrew S Lan, Lawrence Carin, and Richard Baraniuk. Test-size reduction for concept estimation. In Educational Data Mining 2013, 2013.
- [193] Peter Vovsha. The cross-nested logit model: application to mode choice in the Tel-Aviv metropolitan area. Transportation Research Board, 1997.
- [194] Andrew E Waters, Andrew Lan, Christoph Studer, and Richard G Baraniuk. Learning analytics via sparse factor analysis. In Personalizing education with machine learning, nips 2012 workshop, 2012.
- [195] David J Weiss. Improving measurement quality and efficiency with adaptive testing. Applied psychological measurement, 6(4):473–492, 1982.

- [196] David J Weiss and G Kingsbury. Application of computerized adaptive testing to educational problems. Journal of Educational Measurement, 21(4):361–375, 1984.
- [197] Chieh-Hua Wen and Frank S Koppelman. The generalized nested logit model. Transportation Research Part B: Methodological, 35(7):627–641, 2001.
- [198] Ruby C Weng and Chih-Jen Lin. A bayesian approximation method for online ranking. Journal of Machine Learning Research, 12(Jan):267–300, 2011.
- [199] Jacob Whitehill, Ting-fan Wu, Jacob Bergsma, Javier R Movellan, and Paul L Ruvolo. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In Advances in neural information processing systems, pages 2035–2043, 2009.
- [200] Joseph Jay Williams, Juho Kim, Anna Rafferty, Samuel Maldonado, Krzysztof Z Gajos, Walter S Lasecki, and Neil Heffernan. Axis: Generating explanations at scale with learnersourcing and machine learning. In Proceedings of the Third (2016) ACM Conference on Learning@ Scale, pages 379–388. ACM, 2016.
- [201] Suzanne M Wilson and Penelope L Peterson. Theories of learning and teaching: what do they mean for educators? National Education Association Washington, DC, 2006.
- [202] John H Wolfe. Automatic question generation from text-an aid to independent study. ACM SIGCSE Bulletin, 8(1):104–112, 1976.
- [203] William Wu, Constantinos Daskalakis, Nicolaas Kaashoek, Christos Tzamos, and Matthew Weinberg. Game theory based peer grading mechanisms for moocs. 2015.
- [204] Yan Yan, Glenn M Fung, Rómer Rosales, and Jennifer G Dy. Active learning from crowds. In Proceedings of the 28th international conference on machine learning (ICML-11), pages 1161–1168, 2011.
- [205] Xuchen Yao and Yi Zhang. Question generation with minimal recursion semantics. In Proceedings of QG2010: The Third Workshop on Question Generation, pages 68–75. Citeseer, 2010.

- [206] Mark Yatskar, Bo Pang, Cristian Danescu-Niculescu-Mizil, and Lillian Lee. For the sake of simplicity: Unsupervised extraction of lexical simplifications from wikipedia. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pages 365–368. Association for Computational Linguistics, 2010.
- [207] Hezheng Yin, Joseph Moghadam, and Armando Fox. Clustering student programming assignments to multiply instructor leverage. In Proceedings of the Second (2015) ACM Conference on Learning@ Scale, pages 367–372. ACM, 2015.
- [208] Michael Yudelson and Peter Brusilovsky. Navex: Providing navigation support for adaptive browsing of annotated code examples. In AIED, volume 5, pages 710–717, 2005.
- [209] Rick Zahar, Tom Cobb, and Nina Spada. Acquiring vocabulary through reading: Effects of frequency and contextual richness. Canadian Modern Language Review, 57(4):541–572, 2001.
- [210] Martin B Zarrop. Optimal experiment design for dynamic system identification. PhD thesis, Imperial College London (University of London), 1977.
- [211] Ke Zhou, Shuang-Hong Yang, and Hongyuan Zha. Functional matrix factorizations for cold-start recommendation. In Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval, pages 315–324. ACM, 2011.
- [212] Ciyou Zhu, Richard H Byrd, Peihuang Lu, and Jorge Nocedal. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. ACM Transactions on Mathematical Software (TOMS), 23(4):550–560, 1997.
- [213] Zhengyuan Zhu and Michael L Stein. Spatial sampling design for parameter estimation of the covariance function. Journal of Statistical Planning and Inference, 134(2):583–603, 2005.
- [214] Klaus Zwerina, Joel Huber, and Warren F Kuhfeld. A general method for constructing efficient choice designs. Durham, NC: Fuqua School of Business, Duke University, 1996.