Data-Driven Models with Applications in Crowdsourcing, Retail, and Peer-to-Peer Lending

by

Kevin Jiao

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Technology, Operations, and Statistics

New York University

May, 2019

Maxime C. Cohen

Abstract

Making predictions based on historical data has a wide range of applications. Nowadays with the advent of more and more online platforms, we are able to design various machine learning models and test them with online data. In this thesis, there are three main chapters, each focusing on a specific application of data-driven modeling problems. The first chapter is devoted to the development of rank aggregation algorithms in the setting of crowdsourcing. The basic question we want to answer is how to efficiently collect a large amount of high-quality pairwise comparisons for the ranking purpose. The second chapter focuses on designing and comparing different demand prediction models for online retail with promotions. The data obtained from online retail platforms has a distinctive hierarchical structure and our goal is to utilize such a structure to help us achieve higher prediction accuracy. Experimental evaluations on both synthetic and real data are available at the end of the chapter. Finally, the last chapter focuses on the online peer-to-peer lending market. In particular, we want to infer the impact of IPOs on firms operational behaviors. We use one firm (public) as treatment and the other one (private) as control to conduct causal inference.

Acknowledgments

First and foremost, I would like to thank my advisor, Maxime Cohen, for his guidance and mentorship throughout my PhD studies. There are countless occasions when we spent hours together brainstorming interesting ideas and developing new methods. He helped me materialize many of them, which contribute to most of this thesis. Without him, I would not be able to pursue that many fascinating problems in the data-driven realm. Maxime also provided me the opportunity to talk to different companies and make a practical impact while doing rigorous academic research. I was constantly amazed by his creative thinking, ability to convey ideas to different audience, and wide range of knowledge in various subjects. Furthermore, his passion, dedication, and attentions to detail are truly inspirational to me. I have learned so much from him in the past several years and have truly evolved as a person.

My thesis committee consists of Xi Chen, Philip Zhang, and Mike Pinedo. I would like to thank all of them for their valuable feedbacks on my thesis. It was Xi who, right after I joined Stern, introduced me to machine learning and gave me the chance to work on a very interesting project. Thesis aside, I am grateful to have the privilege learning from other collaborators including Qihang Lin, Daniel Guetta, Foster Provost. I would also like to say special thanks to other Stern professors including Jiawei Zhang, Josh Reed, Ilan Lobel, Mor Armony, and Srikanth Jagabathula. It is my honor to have the chance thriving in a community full of world-renowned scholars in the past five years.

Life in New York City would not have been the same without my best friends Jason Tian and Quinn Chen. From roommates to brothers, we bond through many fun social activities. The laughters we share will always be some of my most precious memories. I also remember and appreciate the support from my other friends and colleagues: Sherry Zhou, Tianchen Wei, Justin Xia, Tiffany Chu, Steven Parad, Dmitry Mitrofanov, Yunchao Xu, Tarek Abdallah. Thanks for always being there for me.

Finally, my thanks go to my family. Their love makes me who I am today. In particular, I would like to thank my mother. Without her, I would not be able to get through this journey. This thesis is dedicated to her.

Contents

1 Introduction **12**

2 Crowdsourcing **14**

2.1 Introduction 14

2.2 Related Work 16

2.3 Crowdsourced Ranking by Homogeneous Workers 18

2.3.1 Model Setup 18

2.3.2 Bayesian Decision Process 20

2.3.3 Approximated Knowledge Gradient Policy 23

2.4 Crowdsourced Ranking by Heterogeneous Workers 29

2.4.1 Model Setup 29

2.4.2 Bayesian Decision Process 30

2.4.3 Approximated Knowledge Gradient Policy 32

2.5 Experiment 35

2.5.1 Simulated Study under the Homogeneous Workers Setting 36

2.5.2 Simulated Study under the Heterogeneous Workers Setting 40

2.5.3 Real Data Study 44

2.6 Conclusion 45

2.7 Appendix 47

3 Retail **50**

3.1 Introduction 50

3.1.1 Main Results and Contributions 51

3.1.2 Related Literature 51

3.2 Model 53

3.3 Dynamic Data Aggregation with Clustering 55

3.4 Theoretical Properties of DAC 58

3.5 Simulated Experiments 63

3.5.1 Linear Regression . . . 63
3.5.2 Logistic Regression . . . 67
3.6 Applying DAC to Retail Data . . . 70
3.6.1 Data . . . 70
3.6.2 Prediction Performance . . . 71
3.6.3 Managerial Insights . . . 74
3.7 Conclusion . . . 76
3.8 Appendix . . . 77
3.8.1 Two Potential Methods . . . 77
3.8.2 Proofs of Statements . . . 79

4 Peer-to-Peer Lending **83**
4.1 Introduction . . . 83
4.1.1 Related Literature . . . 86
4.2 Data and Metrics . . . 87
4.2.1 Data . . . 87
4.2.2 Metrics . . . 89
4.3 Econometric Methods . . . 90
4.3.1 Self-Platform Analysis . . . 90
4.3.2 Cross-Platform Analysis . . . 92
4.3.3 Moderating Factor . . . 95
4.4 Results . . . 95
4.4.1 Self-Platform Results . . . 95
4.4.2 Cross-Platform Results . . . 100
4.4.3 Summary of Results for Other Time Events . . . 106
4.5 Insights and Main Findings . . . 107
4.6 Robustness . . . 108
4.6.1 Varying the Number of Months . . . 108
4.6.2 Alternative Risk Level Definition . . . 109
4.6.3 Alternative Return Definitions . . . 110
4.6.4 Alternative Split for Synthetic Control . . . 112
4.7 Conclusion . . . 112
4.8 Appendix . . . 114
4.8.1 Full Regression Tables . . . 114
4.8.2 Diff-in-Diffs Results with Moderating Factor . . . 115
4.8.3 Synthetic Control Results for Risky, Average, and Safe Loans . . . 117
4.8.4 Diff-in-Diffs Results for $\mathbf{r_1 = 0.11}$ and $\mathbf{r_2 = 0.14}$ . . . 120

List of Figures

2.1 Performance comparison under the homogeneous workers setting. The x-axis is the budget level and y-axis is the averaged ranking accuracy. 37
2.2 Heat map of labeling frequency for item pairs with different levels of ambiguity 39
2.3 Heat map of labeling frequency for pairs with very close scores 40
2.4 Performance comparison under the heterogeneous workers setting. The x-axis is the budget level and y-axis is the averaged ranking accuracy. 40
2.5 Density plots of different Beta distributions for generating ρ_w 41
2.6 Comparisons between AKG using Beta(4,1) prior and AKG using the true generating distribution as prior. 42
2.7 Averaged number of comparisons (a.k.a., labeling frequency) made by workers with different levels of reliability ρ_w. 43
2.8 Performance comparison on the real dataset 44

3.1 Comparison of standard errors for the aggregated and decentralized models 60
3.2 Comparison of large estimation error probability for aggregated and decentralized models 62
3.3 Performance comparison of different prediction models (for linear regression) 65
3.4 Performance comparison of different prediction models (for linear regression) 66
3.5 Performance comparison of different prediction models (for logistic regression) 68
3.6 Performance comparison of different prediction models (for logistic regression) 69
3.7 Performance comparison using real data 73
3.8 Per-item prediction accuracy comparison 74

4.1 Comparison of the histograms using original and matched data 102
4.2 Trend visualization of two dependent variables 104
4.3 Loan performance comparison between pre- and post-treatment 104
4.4 Borrowers' attributes comparison between pre- and post-treatment 105
4.5 Comparison of pre- and post-treatment loans 113
4.6 Comparison of pre- and post-treatment for Risky loans 117

4.7 Comparison of pre- and post-treatment for Average loans 118

4.8 Comparison of pre- and post-treatment for Safe loans 119

List of Tables

2.1 Comparison in computation time under the homogeneous workers setting. 38

2.2 Computation time under the heterogeneous workers setting. 42

3.1 Maximum value of d_x 61

3.2 Parameters used in Section 3.5.1 63

3.3 Parameters used in Section 3.5.2 67

3.4 Summary statistics of each department 70

3.5 Fields in our dataset (observations are aggregated at the week-SKU level) 71

3.6 Number of estimated coefficients 75

4.1 Summary statistics for P_1 and P_2 88

4.2 The dependent variables used in this chapter 90

4.3 T-test results for P_1 and P_2 96

4.4 T-test results for P_1 and P_2 (Risky loans) 97

4.5 T-test results for P_1 and P_2 (Average loans) 97

4.6 T-test results for P_1 and P_2 (Safe loans) 97

4.7 Regression results for P_1 and P_2 99

4.8 Regression results for P_1 (with moderator) 99

4.9 Regression results for P_2 (with moderator) 99

4.10 Generic diff-in-diffs results 101

4.11 Diff-in-diffs results with PSM 103

4.12 Generic diff-in-diffs results ($\Delta t = 4$) 109

4.13 Diff-in-diffs results for Risky loans ($r_1 = 0.09, r_2 = 0.16$) 109

4.14 Diff-in-diffs results for Average loans ($r_1 = 0.09, r_2 = 0.16$) 110

4.15 Diff-in-diffs results for Safe loans ($r_1 = 0.09, r_2 = 0.16$) 110

4.16 T-test results for P_1 under different return definitions 111

4.17 T-test results for P_2 under different return definitions 111

4.18 Diff-in-diffs results under different return definitions 111

4.19 Estimated regression coefficients of P_1 for default rate, return, and credit score 114

4.20 Estimated regression coefficients of P_2 for default rate, return, and credit score 114

4.21 Estimated diff-in-diffs coefficients for default rate, return, and credit score 115

4.22 Diff-in-diffs results for Risky loans 115

4.23 Diff-in-diffs results for Average loans 115

4.24 Diff-in-diffs results for Safe loans 116

4.25 Diff-in-diffs results for Risky loans ($r_1 = 0.11, r_2 = 0.14$) 120

4.26 Diff-in-diffs results for Average loans ($r_1 = 0.11, r_2 = 0.14$) 120

4.27 Diff-in-diffs results for Safe loans ($r_1 = 0.11, r_2 = 0.14$) 120

Chapter 1

Introduction

Using data to build models and make predictions has always been an interesting topic in research. In the last several years, practitioners have been more and more fascinated by the development in the field data-driven machine learning. The intersection of prediction modeling and traditional operations management has brought countless opportunities for researcher. Particularly, we have witnessed a rapid development of data analytics and optimization in areas such as retail, advertisement, marketing, and finance. The caveat here is to on one hand, design better models to make accurate predictions, while on the other hand maintain a rigorous theoretical framework and deliver sound business insights. In such fashion, this thesis investigates three specific applications of data-driven modeling.

In Chapter 2, we tackle the problem of online crowdranking. The content is based on paper [21]. Rank aggregation based on pairwise comparisons over a set of items has a wide range of applications. Although considerable research has been devoted to the development of rank aggregation algorithms, one basic question is how to efficiently collect a large amount of high-quality pairwise comparisons for the ranking purpose. Because of the advent of many crowdsourcing services, a crowd of workers are often hired to conduct pairwise comparisons with a small monetary reward for each pair they compare. Since different workers have different levels of reliability and different pairs have different levels of ambiguity, it is desirable to wisely allocate the limited budget for comparisons among the pairs of items and workers so that the global ranking can be accurately inferred from the comparison results. To this end, we model the active sampling problem in *crowdsourced ranking* as a Bayesian Markov decision process, which dynamically selects item pairs and workers to improve the ranking accuracy under a budget constraint. We further develop a computationally efficient sampling policy based on knowledge gradient as well as a moment matching technique for posterior approximation. Experimental evaluations on both synthetic and real data show that the proposed policy achieves high ranking accuracy with a lower labeling cost.

In Chapter 3, we switch to the field of retail operations. Retailers collect large volumes of transaction data with the goal of predicting future demand. High accuracy in demand prediction allows retailers to better manage their inventory, and ultimately mitigate stock-outs and excess supply. A typical retail setting involves predicting demand for hundreds of products simultaneously. While some products have a large amount of historical data, others were recently introduced and transaction data can be scarce. A common approach is to cluster several products together and estimate a joint model at the cluster level. In this vein, one can estimate some model parameters by aggregating the data from several items, and other parameters at the item level. In this chapter, we propose a practical method—referred to as the Dynamic Aggregation with Clustering (DAC) algorithm—that balances the tradeoff between data aggregation and model flexibility. The DAC allows us to predict demand while optimally identifying the features that should be estimated at the (i) item, (ii) cluster, and (iii) aggregated levels. We analytically show that the DAC yields a consistent estimate along with improved asymptotic properties relative to the traditional ordinary least squares method that treats different items in a decentralized fashion. Using both simulated and real data, we illustrate the improvement in prediction accuracy obtained by the DAC relative to several common benchmarks. Interestingly, the DAC not only has theoretical and practical advantages, it also helps retailers discover useful managerial insights.

Finally in Chapter 4, we examine the peer-to-peer lending marketplace. This chapter is based on paper [23]. Peer-to-peer lending platforms have become commonplace for individuals and small firms interested in borrowing capital. Our analysis is based on two large peer-to-peer lending online platforms, which we refer to as P_1 and P_2. We are facing an interesting situation where P_1 went public by filing an initial public offering (IPO), while P_2 remained privately held. Using large loans data from both platforms, we exploit this empirical environment to carefully infer the impact of IPOs on peer-to-peer lending platforms. Did P_1 alter its decisions by accepting different types of loans or by softening its requirements? Or, on the contrary, did its decisions become stricter given the increased level of scrutiny? To answer these questions, we use several econometric tools, including sample selection bias correction, propensity score matching, and synthetic control. We present rigorous empirical analyses at four time events: first IPO rumor, filing date, actual IPO, and quarterly report release. We find that several performance metrics were indeed affected by the IPO filing. Specifically, we observe that (i) the loans' performance (default rate and return) decreased, (ii) borrowers' requirements (credit score and annual income) diminished, and (iii) the acceptance rate inflated.

Chapter 2

Crowdsourcing

2.1 Introduction

Inferring the ranking over a set of items, such as documents, images, movies, or URL links, is an important learning problem with many applications in areas like web search, recommendation systems, online games, etc. An interesting problem related to rank inference is estimating a score for each item based on a certain criterion that the items can be ranked, such as the score of relevance or the score of quality. Typically, both the ranking and the scores of items can be inferred from a collection of high-quality labels on the items. There are mainly two different types of labels. The label of the first type is associated with each individual item in order to characterize the property of the item itself, for example, a binary or an ordinal score (e.g., 5-point grade). The label of the second type is instead associated with a subset of items that reveal their relative properties, for example, a partial ranking that covers only this subset. Labels of both types can be obtained by soliciting the knowledge of human workers, depending on whether the worker is employed to evaluate a single item or to compare a subset of items according to a given criterion. In practice, a binary score usually cannot fully distinguish all items and ordinal scores from different workers are often inconsistent due to the difference in their understandings of the grades in the ordinal scoring scheme. Therefore, the second type of labels has been more widely adopted, which can effectively reduce the impact of misunderstanding among workers and is more appropriate for ranking fine-grained items with a large number of graduations (e.g., in our real data experiment on accessing reading difficulty of an article into one of twelve American grade levels). Moreover, empirical evidences show that the ranking accuracy of a human worker typically decreases when he or she has to compare many items at a time. For this reason, in this chapter, we only consider the relative comparisons over *pairs* of items and the label from a human worker indicates which item is preferred to the other.

The traditional approach of conducting pairwise comparisons by a small group of experts is usually time consuming and expensive. It fails to meet the growing need of labeled data for ranking tasks. Because of the advent of online crowdsourcing services [54] such as Amazon Mechanical Turk, a more efficient and more economic approach has emerged: a large amount of unlabeled pairs of items are posted to a crowdsourcing platform, where a crowd of workers are hired to perform pairwise comparisons and provide labels of the assigned pairs. Given the labels from crowd workers, we can infer a global ranking over all items. We refer to the process of collecting pairwise labels and ranking items as *crowdsourced ranking*.

Despite its availability and scalability, challenges remain in crowdsourced ranking. A certain amount of monetary reward is paid to a worker for each pair of items he or she compares while there is usually only a fixed amount of budget available, limiting the total number of pairwise labels we can collect. Hence, there is a need for a budget-efficient decision process for allocating the budget over item pairs and workers. In particular, on crowdsourcing platforms, there are unreliable workers who submit their answers quickly but carelessly in order to obtain more monetary reward with less effort. Hence, the comparison results provided by crowd workers often contain non-negligible noise. As a remedy, multiple workers are hired to compare the same pair of items independently in the hope that the correct ranking can be recovered, and that the unreliable workers can be identified by comparing their answers with the rest of workers. However, each pairwise comparison will incur a pre-specified monetary cost. Without a careful control, such a repetitive labeling strategy often results in too many labels on the same pair by different workers, leading to a high cost. Furthermore, because of the diversity of their backgrounds and expertise, workers do not always agree with each other in the results of pairwise comparisons, especially when the two items in comparison are competitive to each other. We refer to such a competitive pair as an *ambiguous pair* since the ordering of them is more difficult to be determined. Presumably, a greater budget should be spent on ambiguous pairs, but identifying ambiguous pairs under the budget constraint itself is a challenging problem, which requires some effective learning scheme. Given the trade-off between the labeling cost and the quality of ranking results, there are two fundamental challenges in crowdsourced ranking:

1. Given the inconsistent pairwise labels from crowd workers with different reliability, how to aggregate these labels into a global ranking over items.

2. With both unreliable workers and ambiguous pairs initially unidentified, how to incorporate a learning scheme with an efficient sampling procedure (over both pairs of items and workers) under the budget constraint to achieve the highest ranking accuracy.

To address these challenges, we need to first model the reliability of workers and the ambiguity of item pairs and analyze how they influence the pairwise label. To this end, we adopt a combination of the Bradley-Terry-Luce ranking model [13, 76] for modeling the comparison results and the Dawid-Skene model [32] for workers' reliability. The reason why we adopt the Bradley-Terry-Luce model is that learning such a model will not only provide a ranking over items but also give a score to each item, which can be useful in many applications (e.g., providing player's rating in chess games). We measure the quality of the ranking inferred from the collected labels using the *Kendall's tau rank correlation coefficient* (Kendall's tau for short) with respect to the underlying true ranking.

Under such a model and a quality measure, we propose a dynamic sampling and ranking procedure which addresses the aforementioned two challenges in a unified framework. In particular, we first introduce the priors for items' latent true scores and workers' reliability and formulate the crowdsourced ranking problem into a finite-horizon Bayesian *Markov decision problem* (MDP), whose state variables correspond to the posterior distributions given the observed labels. Here, the number of stages is determined by the total budget, i.e., the total number of pairs that can be requested for labeling. As the budget level increases, the size of the state space grows at an exponential rate, which makes the exact solving of such a MDP problem intractable. To address the computational difficulty, we propose an efficient sampling strategy called *approximated knowledge gradient* (AKG) policy based on the popular knowledge gradient policy [89, 41, 40, 97]. The proposed policy dynamically chooses the next pair of items and the worker that together lead to a maximum expected improvement in Kendall's tau rank correlation coefficient. Finally, to determine the global ranking that maximizes the expected Kendall's tau, one needs to solve a maximum linear ordering problem [48], which is a NP-hard problem (and in fact, APX-hard (approximable-hard) [84]). To address this challenge, we propose a moment matching technique to approximate the posteriors in parametric forms so that the linear ordering problem under the approximated posterior can be easily solved by a simple sorting procedure.

The rest of the chapter is organized as follows. In Section 2.2, we review the related literature. In Section 2.3, we introduce the model and the proposed policy under the simplified case where all workers are homogeneous and perfectly reliable. In Section 2.4, we extend our policy to the case where the crowd workers have heterogeneous reliability. In Section 2.5, we present numerical results on both simulated and real datasets, followed by conclusions in Section 2.6. The detailed proofs and derivations are provided in the appendix.

2.2 Related Work

The dataset of partial rankings over items can be generated from a variety of sources including crowdsourcing services [99], online competition games (e.g., Microsoft's TrueSkill system [52]), and online users' activities such as browsing, clicking and transactions that reveal certain preferences. Learning a global ranking of a large set of items by aggregating a collection of partial rankings/preferences has been an active research area for the past ten years (see, e.g., [46, 85, 117, 98, 99, 94, 75, 108]). However, most work on rank aggregation considers a static estimation problem — inferring a global ranking based on a pre-existing dataset. The problem we consider here is related to but significantly different from these works because we model crowdsourced ranking as a dynamic procedure where the inference of ranking and collection of data proceed concurrently and influence each other.

The crowdsourced ranking problem we considered has a close connection with the dynamic sorting problem using noisy pairwise comparisons, which has been studied by several authors [4, 14, 93, 110, 60]. However, these papers assume the noise of pairwise comparison results has the same distribution for all pairs, which is not reasonable in crowdsourced ranking because workers usually rank significantly different items more correctly than they do for similar items. The approaches proposed by [88] and [91] assume that the labeling noise depends on the latent qualities or features of the items. However, their approaches do not model the reliability of workers in the decision process. In contrast, our approach allows a label's noise to depend not only on the items themselves, but also on the reliability of the worker who provides the label. The ranking model adopted in this chapter, which combines the Bradley-Terry-Luce model and the Dawid-Skene model, was originally proposed in [20], which also considers a similar problem of Bayesian statistical decision-making for crowdsourced ranking. However, the sampling strategy developed in [20], which prioritizes the pair of items and the worker with the highest information gain, is a simple heuristic without a well-defined objective function to be optimized. In contrast, our work chooses the expected Kendall's tau as the objective function to maximize, which guides the development of the knowledge gradient policy.

In addition to crowdsourced ranking, the problem of crowdsourced categorical labeling/classification has been extensively studied in the past five years. Most work aims at solving a static problem, which infers the categorical labels and workers' reliability based on a static problem (see, e.g., [32, 95, 112, 113, 73, 44, 119]). Recently, some research has been devoted to dynamic sampling in crowdsourced classification [64, 63, 7, 35, 61, 53, 22]. In particular, both [61] and [22] utilized the Markov decision process to model the budget allocation (i.e., sampling over items and workers) process. Since we also adopt a Bayesian Markov decision process with a variant of knowledge gradient policy, the spirit of our method is similar to that in [22]. However, since the statistical model for a ranking problem is fundamentally different from that of a classification problem, the Markov decision process in this chapter is significantly different from the one introduced by [22] in many aspects such as the objective function, stage-wise rewards, transition probabilities, optimal policy, etc. For example, the policy by [22] is designed to maximize the expected classification accuracy while our policy aims at maximizing the expected Kendall's tau with respective to the true ranking. In fact, even for a static problem with a given set of collected data, inferring the ranking with the maximum expected Kendall's tau is equivalent to a NP-hard maximum linear ordering problem while classifying items with a maximum expected accuracy can be done in closed-form by Bayesian decision rule. In this chapter, we avoid this computational challenge by exploiting the structure of the expected Kendall's tau and approximating the posteriors using moment matching. We also note that, although one can view the problem of ranking K items as a problem of classifying $K(K-1)/2$ pairs (each pair is treated as an item in [22]), such an approach increases the size of the problem and ignores the dependency between pairwise labels.

In addition, it is worth to note that the problem we consider here is different from the typical tasks in machine-learned ranking or learning to rank [74, 2] where some feature information is available for each item and training data is used to calibrate some statistical models for ranking new items. In contrast to these problems, the feature information is not necessary in our crowdsourced ranking problem. Moreover, besides being applied to ranking items directly, our methods can be utilized to collect training labels for learning to rank problems. According to the type of training data utilized, statistical ranking methods can be classified into three categories [74, 2]: pointwise method, pairwise method and listwise method. The pointwise methods [70, 29, 30] learn a ranking model based on the data of scores or ratings of items. The pairwise methods [42, 16, 120, 17] and the listwise methods [116, 18, 103, 67] learn a ranking model using pairwise comparison results or partial rankings over a subset of items. For the pairwise or listwise methods, the crowdsourced ranking technique we proposed can be used as an upstream procedure that provides high-quality pairwise/listwise comparison data which helps increase the accuracy of the models in the aforementioned papers.

2.3 Crowdsourced Ranking by Homogeneous Workers

In this section, we first consider a simplified setting where workers are homogeneous (we will clarify the meaning of "homogeneous workers" shortly). In Section 2.4, we further extend the developed method for homogeneous workers to heterogeneous workers with different levels of reliability.

2.3.1 Model Setup

We assume that there are K items (denoted by $\{1, \ldots, K\}$) to be ranked and each item i has an *unknown* latent score $\theta_i > 0$ for $i = 1, 2, \ldots, K$. Let $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_K)^T$, where each latent score θ_i models the intensity of preference to item i under some criterion. A *ranking* over K items $\{1, 2, \ldots, K\}$ is a permutation/one-to-one mapping $\pi : \{1, 2, \ldots, K\} \to \{1, 2, \ldots, K\}$ and $\pi(i)$ is the *rank* of item i under π. We follow the convention that $\theta_i > \theta_j$ means item i is preferred to item j and thus item i should have a higher rank than item j. Therefore, the underlying *true ranking* π^* over K items is determined by the ranking of their latent scores, i.e.,

$$\pi^*(i) > \pi^*(j) \quad \text{if and only if} \quad \theta_i > \theta_j. \tag{2.1}$$

We note that the latent scores naturally provide a characterization of *ambiguity for a pair of items*: when the values of θ_i and θ_j are closer, the pair of item i and j is more ambiguous in the sense that the true ordering of them is less obvious.

The way we explore the ranking of θ_i's is through the collection of workers' preferences on different pairs of items. Specifically, we will present only two items at a time to a worker, who will be asked to compare these two items according to the given ranking criterion. Each worker will not be asked to compare the same pair more than once. The results of comparisons will be collected over time and become our historical data, based on which, our task is to infer the true ranking π^*.

In this section, we consider a basic setup where the crowd workers are assumed to be *homogeneous*, meaning that the probabilistic outcomes of their comparisons are only affected by the ambiguities of pairs. More specifically, suppose a worker is randomly selected from the crowd to compare a pair of items i and j with $i < j$ and the comparison result is denoted by a random variable Y_{ij}:

$$Y_{ij} = \begin{cases} 1 & \text{if item } i \text{ is preferred to item } j \text{ by the randomly selected worker} \\ -1 & \text{if item } j \text{ is preferred to item } i \text{ by the randomly selected worker.} \end{cases} \tag{2.2}$$

The setting of homogeneous workers means the probability distribution of Y_{ij} takes the following form

$$\Pr(Y_{ij} = 1) = \frac{\theta_i}{\theta_i + \theta_j} \quad \text{and} \quad \Pr(Y_{ij} = -1) = \frac{\theta_j}{\theta_i + \theta_j} \quad \text{for } i, j = 1, 2, \ldots, K. \tag{2.3}$$

The probabilistic model we used in (2.3) is the well-known Bradley-Terry-Luce (BTL) model [13, 76]. We choose this model for the distribution of Y_{ij} because it admits a simple structure and well fits our framework of dynamic sampling. Furthermore, our method developed for the BTL model can be easily extended to the case of heterogeneous workers which will be studied in Section 2.4.

It is worthwhile to mention that other comparison models can potentially be implemented here. Considering a simplified version of the Thurstone model [105] in which each object i has a score following $N(\theta_i, 1)$, then we have

$$\Pr(Y_{ij} = 1) = \Phi\left(\frac{\theta_i - \theta_j}{\sqrt{2}}\right) \quad \text{and} \quad \Pr(Y_{ij} = -1) = \Phi\left(\frac{\theta_j - \theta_i}{\sqrt{2}}\right).$$

The problem can still be formulated using a Bayesian decision process framework. However, there are several reasons why the BTL model is favored in this chapter. First of all, moment matching under the Thurstone model does not have closed-form solutions and hence we must rely on numerical scheme to compute the first and second moments of the posterior. Second, using moment matching approach, because the posterior is an n-dimensional multivariate Gaussian distribution, we need to update $n(n+1)/2$ parameters (the number of mean parameters plus the number of off-diagonal elements of the covariance matrix) during each iteration of the algorithm whereas with Dirichlet posterior there are only n parameters. Last but not least, with Thurstone model the ranking is no longer a simple sorting of parameters, which is a feature of the BTL model as shown in Theorem 2.

Since each worker can compare the same pair at most once, we assume the size of the crowd workers is large enough so that the distribution of Y_{ij} stays the same after sampling workers without replacement. Note that we can assume $\sum_{i=1}^{K} \theta_i = 1$ without loss of generality since the distribution of Y_{ij} in (2.3) remains unchanged if we multiply each θ_i by the same positive constant. The probability $\frac{\theta_i}{\theta_i+\theta_j}$ in (2.3) can also be interpreted as the percentage of workers in the crowd who prefer item i to item j.

Since the probabilistic model (2.3) does not incorporate or reveal the quality of each worker in the comparison result, in the subsequent study of this section, we only need to focus on how to dynamically select pairs of items to compare. The worker will be selected randomly from the crowd. A dynamic choice over workers will be incorporated into our method in Section 2.4 where the performance of workers is modeled heterogeneously.

2.3.2 Bayesian Decision Process

In a typical crowdsourcing marketplace, a monetary cost must be paid to a worker every time this worker completes a task such as comparing a pair of items. We assume the cost for each comparison is one unit and the total budget available is T units so that at most T pairs (repetition allowed) can be compared in total. Since comparing different pairs will generate different historical data and reveal different information about the true ranking, it is critical to dynamically determine the right sequence of pairs to compare in order to maximize the final ranking accuracy, especially when the budget T is small.

In the traditional offline setting, one needs to determine T pairs at a time beforehand and request the comparisons on those pairs in a batch. The potential problem of such a static approach is that the budget T is not spent in an efficient way to discover the true ranking. In fact, the distribution in (2.3) implies that, when two items have similar latent scores, workers will provide highly inconsistent preferences and it is hard to reach an agreement on such a pair. In this case, the comparison results will be very noisy and one needs to spend more budget on this pair in order to rank them correctly. In contrast, when two items have significantly different latent scores, workers will provide consistent answers so that the additional information we can obtain is little from repeatedly comparing the same two items. In this case, one might want to reduce the budget on such a pair. Unfortunately, without any prior knowledge of the latent scores, it is impossible to decide how much budget should be spent on each pair before observing some comparison results.

In order to efficiently allocate the limited total budget over all pairs, we consider a dynamic crowdsourced ranking policy (Algorithm 1) where only one pair of items is selected and presented to a worker at each time based on historical comparison results. This online method allows the budget to be adaptively shifted towards the ambiguous pairs so that the final ranking accuracy can be improved.

In particular, given the total budget T, the dynamic decision process consists of T stages and, in stage $t = 0, 1, \ldots, T-1$, a pair of items (i_t, j_t) with $i_t < j_t$ is presented to a randomly selected worker and we receive the comparison result $Y_{i_t j_t}$ defined in (2.2) and (2.3). The historical comparison results up to stage t can be summarized by a $K \times K$ matrix M^t with its entry[1] M^t_{ij} equal to the number of times item i is preferred to item j up to stage t. For each stage t where the pair (i_t, j_t) is compared, we define Δ^t to be a sparse $K \times K$ matrix with only one non-zero element: $\Delta^t_{i_t j_t} = 1$ if $Y_{i_t j_t} = 1$ and $\Delta^t_{j_t i_t} = 1$ if $Y_{i_t j_t} = -1$. By its definition, M^t can be updated iteratively as follows

$$M^0 = \mathbf{0}, \quad M^{t+1} = M^t + \Delta^t \quad \text{for } t = 0, 1, \ldots, T-1, \tag{2.4}$$

where $\mathbf{0}$ denotes the $K \times K$ all-zero matrix.

We denote an *adaptive dynamic budget allocation/sampling policy* by $\mathcal{A} = \{(i_t, j_t)\}_{t=0,1,\ldots,T-1}$ where $(i_t, j_t) = (i_t(M^t), j_t(M^t))$ depends on the previous comparison results through M^t. Our goal is to find the best $\mathcal{A}$ so that the inferred ranking based on all the historical comparisons (represented by M^T) achieves

[1] In this chapter, the notation A_{ij} represents the entry in the i-th row and j-th column of matrix A.

the highest accuracy.

To measure the accuracy of an inferred ranking π, we adopt the popular evaluation criterion — normalized *Kendall's tau rank correlation coefficient* [65] between π and π^* (Kendall's tau for short):

$$\begin{aligned} \tau(\pi, \pi^*) &\equiv \frac{|\{(i,j) : i < j,\ \ (\pi(i) - \pi(j))\,(\pi^*(i) - \pi^*(j)) > 0\}|}{K(K-1)/2} \\ &= \frac{2}{K(K-1)} \sum_{i \neq j} \mathbf{1}_{\{\pi(i) > \pi(j)\}} \mathbf{1}_{\{\theta_i > \theta_j\}}, \end{aligned} \tag{2.5}$$

where $\mathbf{1}_{\{\cdot\}}$ denotes the indicator function. Here, the numerator counts the number of pairs that π and π^* agree with each other and the denominator is the total number of pairs over K items. Hence, $\tau(\pi, \pi^*) \in [0, 1]$ and represents the percentage of agreements between π and π^*. The ranking accuracy of π is higher when $\tau(\pi, \pi^*)$ is closer to one and $\pi = \pi^*$ if and only if $\tau(\pi, \pi^*) = 1$.

However, we cannot infer a ranking based on the collected data by directly maximizing $\tau(\pi, \pi^*)$ because π^* and $\boldsymbol{\theta}$ are unknown. To address this challenge, we adopt a Bayesian framework by proposing a prior distribution on $\boldsymbol{\theta}$ and infer a ranking π that maximizes the posterior expectation of $\tau(\pi, \pi^*)$. Recall that the vector of latent scores $\boldsymbol{\theta}$ is assumed to lie in the simplex

$$\Delta \equiv \Big\{ \boldsymbol{\theta} \in \mathbb{R}^k \Big| \sum_{i=1}^{K} \theta_i = 1, \theta_i > 0 \Big\}. \tag{2.6}$$

It is natural to assume that $\boldsymbol{\theta}$ is drawn from a *Dirichlet prior distribution* parameterized by $\boldsymbol{\alpha}^0 = (\alpha_1^0, \ldots, \alpha_K^0)^T$ with $\alpha_i^0 > 0$ for all i (note that Dirichlet distribution of order K is supported on Δ). Namely,

$$\boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha}^0) = \frac{1}{\mathrm{B}(\boldsymbol{\alpha}^0)} \prod_{i=1}^{K} \theta_i^{\alpha_i - 1},$$

where $\mathrm{B}(\boldsymbol{\alpha}) = \frac{\prod_{i=1}^K \Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^K \alpha_i\right)}$ and $\Gamma(x) \equiv \int_0^\infty \lambda^{x-1} e^{-\lambda} \mathrm{d}\lambda$ is the gamma function. Given the comparison data M^t up to stage t and the probability distribution of each comparison result in (2.3), the density function of the posterior distribution of $\boldsymbol{\theta}$ takes the following form,

$$p(\boldsymbol{\theta}|M^t, \boldsymbol{\alpha}^0) = \frac{1}{\mathrm{H}(M^t, \boldsymbol{\alpha}^0)} \prod_{i \neq j} \quad \frac{\theta_i}{\theta_i + \theta_j} \quad {}^{M_{ij}^t} \prod_i \theta_i^{\alpha_i^0 - 1} = \frac{1}{\mathrm{H}(M^t, \boldsymbol{\alpha}^0)} \frac{\prod_{i=1}^K \theta_i^{\beta_i^t + \alpha_i^0 - 1}}{\prod_{i<j} (\theta_i + \theta_j)^{M_{ij}^t + M_{ji}^t}}, \tag{2.7}$$

where $\boldsymbol{\beta}^t = (\beta_1^t, \beta_2^t, \ldots, \beta_K^t)^T$ with $\beta_i^t \equiv \quad {}_{j \neq i} M_{ij}^t$, i.e., the number of times item i is preferred to another item up to stage t, and

$$\mathrm{H}(M^t, \boldsymbol{\alpha}^0) \equiv \int_\Delta \frac{\prod_{i=1}^K \theta_i^{\beta_i^t + \alpha_i^0 - 1}}{\prod_{i<j} (\theta_i + \theta_j)^{M_{ij}^t + M_{ji}^t}} \mathrm{d}\boldsymbol{\theta},$$

is the normalization constant.

With this posterior distribution in place and with M^t at any stage t, we can infer a ranking $\widehat{\pi}_t$ to maximize the posterior expected ranking accuracy measured by its Kendall's tau with respect to π^*, namely, to find

$$\widehat{\pi}_t \quad \in \quad \arg\max_{\pi} \mathbb{E}\left[\tau(\pi, \pi^*) | M^t, \boldsymbol{\alpha}^0\right], \tag{2.8}$$

where the expectation is taken with respect to the posterior distribution $p(\boldsymbol{\theta}|M^t, \boldsymbol{\alpha}^0)$ in (2.7). We denote the corresponding maximum posterior expected accuracy by $h(M^t)$, i.e.,

$$h(M^t) \quad \equiv \quad \max_{\pi} \mathbb{E} \ \ \tau(\pi, \pi^*)|M^t, \boldsymbol{\alpha}^0 \ \ , \tag{2.9}$$

where the dependence of h on the prior $\boldsymbol{\alpha}^0$ is suppressed for notational simplicity. We are interested in finding a dynamic budget allocation policy $\mathcal{A} = \{(i_t, j_t)\}_{t=0,1,\ldots,T-1}$ that maximizes $h(M^T)$, i.e., the final expected ranking accuracy when the budget is exhausted. This problem can be stated as

$$\max_{\mathcal{A}} \mathbb{E}^{\mathcal{A}} \ \ h(M^T)|\boldsymbol{\alpha}^0 \ \ , \tag{2.10}$$

where $\mathbb{E}^{\mathcal{A}}$ represents the expectation over the sample paths (i.e., the sampled pairs and outcomes) generated by the policy $\mathcal{A}$.

The maximization problem in (2.10) can be formulated as a T-stage Bayesian Markov decision process (MDP), where the *state variable* is the posterior distribution in (2.7) or simply the matrix M^t. The *state space* at each stage t denoted by $\mathcal{S}^t$ takes the form of

$$\mathcal{S}^t = \quad M^t \in \mathbb{Z}^{K\times K}_{\geq 0} : \quad \underset{i,j}{} \ M^t_{ij} = t \ \ , \tag{2.11}$$

where $\mathbb{Z}_{\geq 0}$ denotes the set of non-negative integers. The state variable makes a transition according to (2.4) given the observed comparison result $Y_{i_t j_t}$, where the sampled pair (i_t, j_t) is determined by the policy $\mathcal{A}$. The expected transition probabilities take the form of,

$$\mathbb{E} \ \ \Pr(Y_{ij} = 1)|M^t, \boldsymbol{\alpha}^0 \quad = \quad \mathbb{E}\left[\frac{\theta_i}{\theta_i + \theta_j} \ \ M^t, \boldsymbol{\alpha}^0\right] \tag{2.12}$$

$$\mathbb{E} \ \ \Pr(Y_{ij} = -1)|M^t, \boldsymbol{\alpha}^0 \quad = \quad \mathbb{E}\left[\frac{\theta_j}{\theta_i + \theta_j} \ \ M^t, \boldsymbol{\alpha}^0\right] \tag{2.13}$$

for $1 \leq i < j \leq K$ and the expectation is taken over the posterior of $\boldsymbol{\theta}$ in (2.7). To complete the definition of our Bayesian MDP for crowdsourced ranking, we still need to define the *stage-wise reward.* To this end, we rewrite $h(M^T)$ in (2.10) as a telescopic sum,

$$h(M^T) = \underset{t=0,1,\ldots,T-1}{} \ R(M^t, i_t, j_t, Y_{i_t j_t}); \qquad R(M^t, i_t, j_t, Y_{i_t j_t}) \equiv h(M^{t+1}) - h(M^t), \tag{2.14}$$

and note that $R(M^t, i_t, j_t, Y_{i_t j_t}) = h(M^{t+1}) - h(M^t)$ only depends on M^t, i_t, j_t, $Y_{i_t j_t}$. Given (2.14), the maximization problem (2.10) is equivalent to

$$\begin{aligned} & \max_{\mathcal{A}} \mathbb{E}^{\mathcal{A}} \left[h(M^0) + \underset{t=0}{\overset{T-1}{}} \ R(M^t, i_t, j_t, Y_{i_t j_t}) \ \boldsymbol{\alpha}^0 \right] \qquad (2.15) \\ = \quad & h(M^0) + \max_{\mathcal{A}} \mathbb{E}^{\mathcal{A}} \left[\underset{t=0}{\overset{T-1}{}} \ \mathbb{E} \ \ R(M^t, i_t, j_t, Y_{i_t j_t}) \ M^t, \boldsymbol{\alpha}^0 \quad \boldsymbol{\alpha}^0 \right]. \end{aligned}$$

From (2.15), it is clear that $R(M^t, i_t, j_t, Y_{i_t j_t})$ is the *stage-wise reward,* which can be interpreted as the improvement of the expected ranking accuracy after receiving the comparison result $Y_{i_t j_t}$ at stage t for $t = 0, 1, \ldots, T - 1$.

Given the Bayesian MDP in place, we can apply the dynamic programming (DP) algorithm (a.k.a. backward induction) [90] to compute the optimal policy. Although DP finds the optimal policy, its computation is intractable because:

1. The sophisticated form of the posterior distribution in (2.7) makes it difficult to evaluate the posterior expected ranking accuracy $\mathbb{E}\ \ \tau(\pi, \pi^*)|M^t, \boldsymbol{\alpha}^0$ in (2.9) and the expected transition probabilities in (2.12) and (2.13).

2. The maximization problem (2.9) for solving the optimal posterior expected ranking accuracy is essentially a linear ordering problem [48], which is NP-hard in general (see Section 2.3.3 for more details).

3. The size of the state space $\mathcal{S}^t$ grows exponentially in t according to (2.11), which is known as the curse of dimensionality that prevents us from solving (2.15) exactly with the standard techniques such value iteration, policy iteration and linear programming.

To address these challenges, we propose an approximated knowledge gradient policy (AKG) in the next Section.

2.3.3 Approximated Knowledge Gradient Policy

In this section, we describe an approximated policy to solve (2.10), which is computationally efficient and still provides an inferred ranking with high quality. The proposed approximation policy belongs to the family of *knowledge gradient* (KG) policies [49, 40, 89, 97], which is essentially a single-step look-ahead policy. In our problem, the KG policy will sample the next pair of items with the highest expected stage-wise reward in each stage, i.e., choosing the pair (i_t, j_t) such that

$$\begin{aligned} (i_t, j_t) \quad & \in \arg\max\nolimits_{i<j} \quad \mathbb{E}\ \ R(M^t, i_t, j_t, Y_{i_t j_t})\ M^t, \boldsymbol{\alpha}^0 \\ & = \arg\max\nolimits_{i<j} \quad \mathbb{E}\ \ \Pr(Y_{ij} = 1)|M^t, \boldsymbol{\alpha}^0\ \ R(M^t, i_t, j_t, 1) \\ & \qquad\qquad\qquad +\mathbb{E}\ \ \Pr(Y_{ij} = -1)|M^t, \boldsymbol{\alpha}^0\ \ R(M^t, i_t, j_t, -1). \end{aligned} \tag{2.16}$$

Despite its simplicity and wide applicability, the implementation of the KG policy for our problem in (2.16) is still computationally intractable since we have to evaluate the expected stage-wise reward $\mathbb{E}\ \ R(M^t, i_t, j_t, Y_{i_t j_t})\ M^t, \boldsymbol{\alpha}^0$, where two main challenges will arise.

First, we have to evaluate the transition probabilities (2.12) and (2.13) as well as the ranking accuracy (2.9), which can be written as

$$\begin{aligned} h(M^t) \quad & = \quad \max_{\pi} \mathbb{E}\ \ \tau(\pi, \pi^*)|M^t, \boldsymbol{\alpha}^0 \\ & = \quad \max_{\pi} \frac{2 \quad {}_{i/}\ \ \mathbb{E}\ \ \mathbf{1}_{\{\pi(i)>\pi(j)\}} \mathbf{1}_{\{\theta_i>\theta_j\}} |M^t, \boldsymbol{\alpha}^0}{K(K-1)} \\ & = \quad \max_{\pi} \frac{2 \quad {}_{i\neq j}\ \mathbf{1}_{\pi(i)>\pi(j)} \Pr\left(\theta_i > \theta_j | M^t, \boldsymbol{\alpha}^0\right)}{K(K-1)}. \end{aligned} \tag{2.17}$$

However, due to the complicated structure of the posterior distribution $p(\boldsymbol{\theta}|M^t, \boldsymbol{\alpha}^0)$ in (2.7), the expected transition probabilities (2.12) and (2.13) and the posterior probability $\Pr\left(\theta_i > \theta_j | M^t, \boldsymbol{\alpha}^0\right)$ in (2.17) do not admit a closed form so that one needs to use multidimensional numerical integral or sampling techniques to compute their values. Note that for each stage t, we need to evaluate (2.12), (2.13) and $\Pr\left(\theta_i > \theta_j | M^t, \boldsymbol{\alpha}^0\right)$ for all $K(K-1)/2$ pairs. When these quantities cannot be easily computed, the overall computational cost will be extremely expensive.

Second, even if the posterior probabilities $\Pr\left(\theta_i > \theta_j | M^t, \boldsymbol{\alpha}^0\right)$ for all pairs are given, the maximization problem (2.17) with respect to a global ranking π is still very challenging. In fact, this problem is equivalent to the *maximum linear ordering problem (MAX-LOP)* described as follows. Let $G = (V, E, w)$ be a completed directed graph defined on a set V of K nodes, where the edge set E contains the directed arcs between all pairs of nodes and $w(i, j)$ refers to the weight associated with the arc from node i to node j. A tournament D is a sub-graph of G such that, for any pair of nodes i and j, D contains either the arc from i to j or the arc from j to i but not both. The MAX-LOP aims to find an acyclic tournament D with a maximum total weight on its arcs. If we interpret the arc from node i and node j as the preference of node i to node j under a ranking criterion, each acyclic tournament in G corresponds one-to-one to a global ranking of the nodes. Hence, MAX-LOP is equivalent to finding a ranking π such that the total weight $_{\pi(i)>\pi(j)} w(i, j)$ is maximized. In problem (2.17), the nodes correspond to the K items and the weight $w(i, j) = \Pr\left(\theta_i > \theta_j | M^t, \boldsymbol{\alpha}^0\right)$. Unfortunately, the MAX-LOP is known to be a NP-hard problem and in fact, APX (approximable)-complete and thus no PTAS (Polynomial Time Approximation Scheme) under P $\neq$ NP [84].

Given these two challenges, evaluating $\mathbb{E}\ \ R(M^t, i_t, j_t, Y_{i_t j_t})\ M^t, \boldsymbol{\alpha}^0$ and solving (2.16) repeatedly at each stage are computationally intractable. To address this problem, we propose an *approximated knowledge gradient* (AKG) policy, which first replaces the stage-wise reward (2.14) by an approximated but computable reward and then chooses the pair that maximizes this approximated reward. Our approximation scheme starts with approximating the posterior distribution $p(\boldsymbol{\theta}|M^t, \boldsymbol{\alpha}^0)$ in (2.7) recursively using a sequence of Dirichlet distributions $\mathrm{Dir}(\boldsymbol{\alpha}^t)$ for $t = 1, 2, \ldots, T$ based on *moment matching*. One key benefit of such an approximation is that, at each stage t, the approximated posterior distribution of $\boldsymbol{\theta}$ is still a Dirichlet distribution so that the NP-hard MAX-LOP problem in (2.17) will admit a simple solution via a sorting procedure (see Theorem 2).

Although there exist other methods for posterior approximation, these methods cannot be implemented as efficiently as moment matching in our application. For example, some methods such as variational inference (e.g., [8, 86]) minimize the KL-divergence between the exact posterior and the variational posterior, which requires an iterative optimization algorithm as a subroutine. Other methods like Gibbs sampler are computationally expensive in our case because the full conditional distribution does not have a closed form to allow easy sampling. In contrast, the proposed (algorithmic) moment matching admits a closed-form solution for approximating the posterior, which is computationally very efficient, and further provides a

Dirichlet distribution as the approximated posterior, which facilitates solving the MAX-LOP. We note that the close-form update is critical for online crowdsourcing applications to reduce the computation time between two stages. In practice, since the crowd workers want to maximize their return in a short period of time, they may quit the current task if we let them wait for too long before we determine the next pair. Finally, we note that, although providing the theoretical guarantee for such an iterative approximation is hard in the Bayesian setup, we empirically show that the resulting AKG policy will generate a final ranking of a high accuracy with the limited budget.

Now we formally introduce the posterior approximation and AKG policy. Suppose $\boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha})$ for some parameters $\boldsymbol{\alpha} \in \mathbb{R}^K$. We consider a basic case where only one comparison result Y_{ij} for a pair (i, j) with $i < j$ has been observed. In this case, we approximate the posterior $p(\boldsymbol{\theta}|Y_{ij}, \boldsymbol{\alpha})$ by another Dirichlet distribution $\mathrm{Dir}(\boldsymbol{\alpha}')$ such that

$$\begin{aligned} \mathbb{E}\left[\theta_k|\boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha}')\right] &= \mathbb{E}[\theta_k|Y_{ij}, \boldsymbol{\alpha}] \text{ for } k = 1, 2, \dots, K & (2.18)\\ \mathbb{E}\left(\sum_{k=1}^{K} \theta_k^2|\boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha}')\right) &= \mathbb{E}\left(\sum_{k=1}^{K} \theta_k^2|Y_{ij}, \boldsymbol{\alpha}\right). & (2.19) \end{aligned}$$

This system of equations has the following explicit characterization.

Proposition 1. *Suppose $\boldsymbol{\theta} \sim Dir(\boldsymbol{\alpha})$ and Y_{ij} is the only comparison result for $i < j$. Let $\alpha_0 = \sum_{k=1}^{K} \alpha_k$ and $\alpha_0' = \sum_{k=1}^{K} \alpha_k'$. The equations* (2.18) *and* (2.19) *can be represented as*

$$\begin{cases} \frac{\alpha_i'}{\alpha_0'} = \frac{\left(\alpha_i + \frac{1+Y_{ij}}{2}\right)(\alpha_i+\alpha_j)}{\alpha_0(\alpha_i+\alpha_j+1)} \\ \frac{\alpha_j'}{\alpha_0'} = \frac{\left(\alpha_j + \frac{1-Y_{ij}}{2}\right)(\alpha_i+\alpha_j)}{\alpha_0(\alpha_i+\alpha_j+1)} \\ \frac{\alpha_k'}{\alpha_0'} = \frac{\alpha_k}{\alpha_0} \quad \text{for} \quad k = i, j \\ \sum_{k=1}^{K} \frac{\alpha_k'(\alpha_k'+1)}{\alpha_0'(\alpha_0'+1)} = \frac{\left(\alpha_i + \frac{1+Y_{ij}}{2}\right)\left(\alpha_i + \frac{3+Y_{ij}}{2}\right)(\alpha_i+\alpha_j)}{\alpha_0(\alpha_0+1)(\alpha_i+\alpha_j+2)} \\ \qquad + \frac{\left(\alpha_j + \frac{1-Y_{ij}}{2}\right)\left(\alpha_j + \frac{3-Y_{ij}}{2}\right)(\alpha_i+\alpha_j)}{\alpha_0(\alpha_0+1)(\alpha_i+\alpha_j+2)} + \sum_{k \neq i,j} \frac{\alpha_k(\alpha_k+1)}{\alpha_0(\alpha_0+1)}. \end{cases} \tag{2.20}$$

The proof of Proposition 1 is provided in the Appendix. We denote any $\boldsymbol{\alpha}'$ that satisfies (2.18) and (2.19), and thus (2.20), by

$$\boldsymbol{\alpha}' = \mathbf{MM}(\boldsymbol{\alpha}, i, j, Y_{ij}). \tag{2.21}$$

Note that, given $\boldsymbol{\alpha}$, i, j and Y_{ij}, the right-hand sides of (2.20) are all constants so that we can solve $\boldsymbol{\alpha}' = \mathbf{MM}(\boldsymbol{\alpha}, i, j, Y_{ij})$ in a closed form. In fact, we denote the constants on the right hand sides of (2.20) as C_i, C_j, C_k (for $k = i, j$) and D, respectively. It is easy to show that $\sum_{k=1}^{K} C_k = 1$. The first three equalities in (2.20) imply that $\alpha_k' = C_k \alpha_0'$ for $k = 1, 2, \dots, K$ so that the fourth equality in (2.20) can be represented as $\sum_{k=1}^{K} C_k(C_k\alpha_0' + 1) = D(\alpha_0' + 1)$. Solving α_0' from this equation leads to a closed-form for $\boldsymbol{\alpha}' = \mathbf{MM}(\boldsymbol{\alpha}, i, j, Y_{ij})$ as follows

$$\alpha_0' = \frac{D-1}{\sum_{k=1}^{K} C_k^2 - D} \quad \text{and} \quad \alpha_k' = C_k \alpha_0' \text{ for } k = 1, 2, \dots, K. \tag{2.22}$$

Although the above approximation scheme is established for only one comparison result, it produces a Dirichlet distribution $\mathrm{Dir}(\boldsymbol{\alpha}')$ which has the same type as the prior distribution $\mathrm{Dir}(\boldsymbol{\alpha})$. Therefore, as more comparison results are generated sequentially, we can apply this approximation scheme iteratively after each comparison result. In particular, given a policy $\mathcal{A} = \{(i_t, j_t)\}_{t=0,1,\ldots,T-1}$ with $i_t < j_t$ and the comparison results $\{Y_{i_t j_t}\}_{t=0,1,\ldots,T-1}$, we define $\boldsymbol{\alpha}^t$ recursively as

$$\boldsymbol{\alpha}^{t+1} = \mathbf{MM}(\boldsymbol{\alpha}^t, i_t, j_t, Y_{i_t j_t}) \tag{2.23}$$

for $t = 1, 2, \ldots, T$. By doing so, we approximate the posterior distribution $p(\boldsymbol{\theta}|M^t, \boldsymbol{\alpha}^0)$ by the Dirichlet distribution $\mathrm{Dir}(\boldsymbol{\alpha}^t)$ for $t = 1, 2, \ldots, T$.

With $p(\boldsymbol{\theta}|M^t, \boldsymbol{\alpha}^0)$ approximated by $\mathrm{Dir}(\boldsymbol{\alpha}^t)$, we can mitigate the two challenges mentioned at the beginning of this subsection. First, we can approximate (2.12) and (2.13) as

$$\begin{aligned} \mathbb{E}\ \Pr(Y_{ij} = 1)\ M^t, \boldsymbol{\alpha}^0 \ &\approx \mathbb{E}\ \ \frac{\theta_i}{\theta_i + \theta_j}\ \boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha}^t) \quad = \quad \frac{\alpha_i^t}{\alpha_i^t + \alpha_j^t} && (2.24) \\ \mathbb{E}\ \Pr(Y_{ij} = -1)\ M^t, \boldsymbol{\alpha}^0 \ &\approx \mathbb{E}\ \ \frac{\theta_i}{\theta_i + \theta_j}\ \boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha}^t) \quad = \quad \frac{\alpha_j^t}{\alpha_i^t + \alpha_j^t} && (2.25) \end{aligned}$$

and approximate $\Pr\left(\theta_i > \theta_j|M^t, \boldsymbol{\alpha}^0\right)$ in (2.7) as

$$\Pr\left(\theta_i > \theta_j|M^t, \boldsymbol{\alpha}^0\right) \approx \Pr\left(\theta_i > \theta_j|\boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha}^t)\right) = {}_{\frac{1}{2}}^{1}\ t^{\alpha_i^t - 1}(1-t)^{\alpha_j^t - 1}\mathrm{d}t = I_{\frac{1}{2}}(\alpha_j^t, \alpha_i^t), \tag{2.26}$$

where $I_x(a, b) = \dfrac{\mathrm{B}(x;\ a, b)}{\mathrm{B}(a, b)}$ is known as the *regularized incomplete beta function* with $\mathrm{B}(x;\ a, b) = {}_0^x\ \lambda^{a-1}\,(1-\lambda)^{b-1}\,\mathrm{d}\lambda$ and $\mathrm{B}(a, b) = {}_0^1\ \lambda^{a-1}(1-\lambda)^{b-1}\,\mathrm{d}\lambda$. Note that the approximated quantities in (2.24), (2.25) and (2.26) are much easier to compute than the original ones.

More importantly, the approximation (2.26) simplifies the NP-hard MAX-LOP in (2.17):

$$\max_{\pi} \mathbb{E}\ \ \tau(\pi, \pi^*)|M^t, \boldsymbol{\alpha}^0 \ \approx \max_{\pi} \mathbb{E}\ \ \tau(\pi, \pi^*)|\boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha}^t)\ \ .$$

The right-hand side is still a MAX-LOP but has a special structure so that it can be solved easily by a simple sorting procedure. In particular, the following theorem shows that when $\boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha})$, the optimal ranking in (2.16) can be obtained by sorting the components of $\boldsymbol{\alpha}$.

Theorem 2. *Suppose $\boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha})$. We have*

$$\begin{aligned} \Pi_{\boldsymbol{\alpha}} \ &\equiv\ \{\pi|\pi \textit{ is a ranking of } \{1, 2, \ldots, K\} \textit{ such that } \pi(i) > \pi(j) \textit{ only if } \alpha_i \geq \alpha_j \textit{ for all } i, j\} \\ &=\ \arg\max_{\pi} \mathbb{E}\left[\tau(\pi, \pi^*)|\boldsymbol{\theta} \sim \mathit{Dir}(\boldsymbol{\alpha})\right] && (2.27) \end{aligned}$$

Proof. We first show that $\arg\max_{\pi} \mathbb{E}\left[\tau(\pi, \pi^*)|\boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha})\right] \subset \Pi_{\boldsymbol{\alpha}}$. Suppose $\hat{\pi}$ is the optimal solution of (2.27) where $\hat{\pi}(j) > \hat{\pi}(i)$ for a pair i and j with $\alpha_i > \alpha_j$. We put all items in a row with their ranks given by $\hat{\pi}$ decreasing from the left to the right and obtain a pattern like

$$X \cdots X j \underbrace{X \cdots X}_{S} i X \cdots X,$$

where X represents some item different from i and j and S represents the set of items ranked between i and j. We will show that the objective value of (2.27) can be increased by switching the ranks of i and j.

Recall that the expected accuracy of $\hat{\pi}$ can be represented as

$$\begin{aligned}\mathbb{E}\left[\tau(\hat{\pi},\pi^*)|\boldsymbol{\theta}\sim \mathrm{Dir}(\boldsymbol{\alpha})\right] &= \frac{2}{K(K-1)} \mathop{}_{i\ \neq j} \mathbf{1}_{\hat{\pi}(i\ )>\hat{\pi}(j\ )}\mathrm{Pr}\left(\theta_{i}\ >\theta_{j}\ |\boldsymbol{\theta}\sim \mathrm{Dir}(\boldsymbol{\alpha})\right) \qquad (2.28)\\ &= \frac{2}{K(K-1)}\ \ I_{\frac{1}{2}}(\alpha_i,\alpha_j) + \mathop{}_{s\in S} I_{\frac{1}{2}}(\alpha_s,\alpha_j) + \mathop{}_{s\in S} I_{\frac{1}{2}}(\alpha_i,\alpha_s) + C\ \ ,\end{aligned}$$

where C is the summation of the remaining terms like $I_{\frac{1}{2}}(\alpha_{i}\ ,\alpha_{j}\ )$ which have either at least one of i' and j' not in $S\cup\{i,j\}$ or both i' and j' in S.

Note that switching the ranks of i and j does not change the values of the terms in C. In fact, after such a switch, we obtain a new ranking $\hat{\pi}'$ whose objective value in (2.27) is

$$\mathbb{E}\left[\tau(\hat{\pi}',\pi^*)|\boldsymbol{\theta}\sim \mathrm{Dir}(\boldsymbol{\alpha})\right] = \frac{2}{K(K-1)}\ \ I_{\frac{1}{2}}(\alpha_j,\alpha_i) + \mathop{}_{s\in S} I_{\frac{1}{2}}(\alpha_j,\alpha_s) + \mathop{}_{s\in S} I_{\frac{1}{2}}(\alpha_s,\alpha_i) + C\ \ .$$

Using the fact that $I_{\frac{1}{2}}(a,b)$ is monotonically decreasing in a and monotonically increasing in b and noticing that $\alpha_i > \alpha_j$, we have

$$I_{\frac{1}{2}}(\alpha_j,\alpha_i) + \mathop{}_{s\in S} I_{\frac{1}{2}}(\alpha_j,\alpha_s) + \mathop{}_{s\in S} I_{\frac{1}{2}}(\alpha_s,\alpha_i) > I_{\frac{1}{2}}(\alpha_i,\alpha_j) + \mathop{}_{s\in S} I_{\frac{1}{2}}(\alpha_s,\alpha_j) + \mathop{}_{s\in S} I_{\frac{1}{2}}(\alpha_i,\alpha_s),$$

which implies $\mathbb{E}\left[\tau(\hat{\pi}',\pi^*)|\boldsymbol{\theta}\sim \mathrm{Dir}(\boldsymbol{\alpha})\right] > \mathbb{E}\left[\tau(\hat{\pi},\pi^*)|\boldsymbol{\theta}\sim \mathrm{Dir}(\boldsymbol{\alpha})\right]$, contradicting with the optimality of $\hat{\pi}$. Hence, we can have $\hat{\pi}(i) > \hat{\pi}(j)$ only if $\alpha_i \geq \alpha_j$, meaning that $\hat{\pi}\in\Pi_{\boldsymbol{\alpha}}$.

We then show $\arg\max_{\pi}\mathbb{E}\left[\tau(\pi,\pi^*)|\boldsymbol{\theta}\sim \mathrm{Dir}(\boldsymbol{\alpha})\right] = \Pi_{\boldsymbol{\alpha}}$ by showing that $\mathbb{E}\left[\tau(\pi,\pi^*)|\boldsymbol{\theta}\sim \mathrm{Dir}(\boldsymbol{\alpha})\right]$ has the same value for any $\pi\in\Pi_{\boldsymbol{\alpha}}$. Suppose $\hat{\pi}$ and $\hat{\pi}'$ both belong to $\Pi_{\boldsymbol{\alpha}}$ and there exists a pair i and j with $i = j$ such that $\hat{\pi}(i) > \hat{\pi}(j)$ and $\hat{\pi}'(j) > \hat{\pi}'(i)$. By the definition of $\Pi_{\boldsymbol{\alpha}}$, we have $\alpha_i = \alpha_j$ so that

$$\mathrm{Pr}\left(\theta_i > \theta_j|\boldsymbol{\theta}\sim \mathrm{Dir}(\boldsymbol{\alpha})\right) = I_{\frac{1}{2}}(\alpha_j,\alpha_i) = \frac{1}{2} = I_{\frac{1}{2}}(\alpha_i,\alpha_j) = \mathrm{Pr}\left(\theta_j > \theta_i|\boldsymbol{\theta}\sim \mathrm{Dir}(\boldsymbol{\alpha})\right).$$

This means

$$\begin{aligned}&\mathbf{1}_{\hat{\pi}(i)>\hat{\pi}(j)}\mathrm{Pr}\left(\theta_i > \theta_j|\boldsymbol{\theta}\sim \mathrm{Dir}(\boldsymbol{\alpha})\right) + \mathbf{1}_{\hat{\pi}(j)>\hat{\pi}(i)}\mathrm{Pr}\left(\theta_j > \theta_i|\boldsymbol{\theta}\sim \mathrm{Dir}(\boldsymbol{\alpha})\right)\\ =\ &\mathbf{1}_{\hat{\pi}\ (i)>\hat{\pi}\ (j)}\mathrm{Pr}\left(\theta_i > \theta_j|\boldsymbol{\theta}\sim \mathrm{Dir}(\boldsymbol{\alpha})\right) + \mathbf{1}_{\hat{\pi}\ (j)>\hat{\pi}\ (i)}\mathrm{Pr}\left(\theta_j > \theta_i|\boldsymbol{\theta}\sim \mathrm{Dir}(\boldsymbol{\alpha})\right)\end{aligned}$$

for any pair i and j so that $\mathbb{E}\left[\tau(\hat{\pi},\pi^*)|\boldsymbol{\theta}\sim \mathrm{Dir}(\boldsymbol{\alpha})\right] = \mathbb{E}\left[\tau(\hat{\pi}',\pi^*)|\boldsymbol{\theta}\sim \mathrm{Dir}(\boldsymbol{\alpha})\right]$ by the formulation (2.28), which completes the proof. □

Given a parameter vector $\boldsymbol{\alpha}$, we denote any ranking in $\Pi_{\boldsymbol{\alpha}}$ by $\pi_{\boldsymbol{\alpha}}$. Using moment matching and Theorem

Algorithm 1 Approximated Knowledge Gradient Policy with Homogeneous Workers

```
Initialization: Choose $\boldsymbol{\alpha}^0$ for the prior distribution. Let $M^0$ be a $K \times K$ all-zero matrix.
For $t = 0, \dots, T-1$ do
  1: For each pair $(i, j)$ with $i < j$, compute $\tilde{R}(\boldsymbol{\alpha}^t, i, j, 1)$ and $\tilde{R}(\boldsymbol{\alpha}^t, i, j, -1)$ according to (2.29).
  2: Select $(i_t, j_t)$ such that
       $$(i_t, j_t) \in \arg\max_{i<j} \ \frac{\alpha_i^t}{\alpha_i^t + \alpha_j^t}\tilde{R}(\boldsymbol{\alpha}^t, i, j, 1) + \frac{\alpha_i^t}{\alpha_i^t + \alpha_j^t}\tilde{R}(\boldsymbol{\alpha}^t, i, j, -1) \qquad (2.31)$$
     and present item $i_t$ and item $j_t$ to a randomly selected worker and receive the comparison result $Y_{i_t j_t}$.
  3: According to (2.21) and (2.22), compute
       $$\boldsymbol{\alpha}^{t+1} = \mathbf{MM}(\boldsymbol{\alpha}^t, i_t, j_t, Y_{i_t j_t}) \qquad (2.32)$$
End For
Return: The aggregated ranking $\pi_{\boldsymbol{\alpha}^T}$ obtained by sorting the components of $\boldsymbol{\alpha}^T$.
```

2, we can approximate the stage-wise reward $R(M^t, i, j, Y_{ij})$ by

$$\begin{aligned} R(M^t, i, j, Y_{ij}) &= h(M^{t+1}) - h(M^t) \\ &= \max_{\pi} \mathbb{E} \ \ \tau(\pi, \pi^*)|M^{t+1}, \boldsymbol{\alpha}^0 \ \ - \max_{\pi} \mathbb{E} \ \ \tau(\pi, \pi^*)|M^t, \boldsymbol{\alpha}^0 \\ &\approx \max_{\pi} \mathbb{E}\left[\tau(\pi, \pi^*)|\boldsymbol{\theta} \sim \mathrm{Dir}(\hat{\boldsymbol{\alpha}})\right] - \max_{\pi} \mathbb{E} \ \ \tau(\pi, \pi^*)|\boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha}^t) \\ &= \mathbb{E}\left[\tau(\pi_{\hat{\boldsymbol{\alpha}}}, \pi^*)|\boldsymbol{\theta} \sim \mathrm{Dir}(\hat{\boldsymbol{\alpha}})\right] - \mathbb{E} \ \ \tau(\pi_{\hat{\boldsymbol{\alpha}}}, \pi^*)|\boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha}^t) \\ &= \frac{2}{K(K-1)}\left(\qquad\qquad\qquad\qquad I_{\frac{1}{2}}(\hat{\alpha}_{j} \ , \hat{\alpha}_{i} \ ) - \qquad\qquad\qquad\qquad I_{\frac{1}{2}}(\alpha_{j}^t \ , \alpha_{i}^t \ )\right)_{i \ ,j \ \ : \ \pi_{\hat{\boldsymbol{\alpha}}}(i \ )>\pi_{\hat{\boldsymbol{\alpha}}}(j \ ) \qquad i \ ,j \ \ : \ \pi_{\boldsymbol{\alpha}^t}(i \ )>\pi_{\boldsymbol{\alpha}^t}(j \ )} \\ &\equiv \tilde{R}(\boldsymbol{\alpha}^t, i, j, Y_{ij}) \qquad (2.29) \end{aligned}$$

where $\hat{\boldsymbol{\alpha}} = \mathbf{MM}(\boldsymbol{\alpha}^t, i, j, Y_{ij})$, the third equality is from Theorem 2 and the fourth equality is due to (2.26). Putting (2.16), (2.24), (2.25), and (2.29) together, we can approximate the expected stage-wise reward $\mathbb{E} \ \ R(M^t, i, j, Y_{ij}) \ \ M^t, \boldsymbol{\alpha}^0$ as

$$\begin{aligned} & \mathbb{E} \ \ R(M^t, i, j, Y_{ij}) \ \ M^t, \boldsymbol{\alpha}^0 \\ = \ & \mathbb{E} \ \ \Pr(Y_{ij} = 1)|M^t, \boldsymbol{\alpha}^0 \ \ R(\boldsymbol{\alpha}^t, i, j, 1) + \mathbb{E} \ \ \Pr(Y_{ij} = -1)|M^t, \boldsymbol{\alpha}^0 \ \ R(M^t, i, j, -1) \\ \approx \ & \frac{\alpha_i^t}{\alpha_i^t + \alpha_j^t}\tilde{R}(\boldsymbol{\alpha}^t, i, j, 1) + \frac{\alpha_i^t}{\alpha_i^t + \alpha_j^t}\tilde{R}(\boldsymbol{\alpha}^t, i, j, -1). \qquad (2.30) \end{aligned}$$

The proposed AKG policy will choose the pair (i_t, j_t) that maximizes the approximated expected stage-wise reward in (2.30). As a summary, we describe the AKG policy as Algorithm 1.

It is noteworthy that it is easy to implement a *batch version* of Algorithm 1. In fact, the AKG policy in Algorithm 1 is known as an *index policy* where the right-hand side of (2.31), which calculates the marginal improvement on the ranking accuracy, can be treated as the index for each pair of items. The AKG policy selects the pair with the highest index at each stage. In the batch version, instead of selecting only one pair, one heuristics is to select the top B pairs and distribute to workers simultaneously, where B is a pre-defined batch size. Such a batch implementation can reduce the waiting time of crowd workers and thus accelerate the ranking procedure. Moreover, the AKG policy can be combined with some other batch optimization techniques [115] to determine the optimal set of pairs to evaluate next.

2.4 Crowdsourced Ranking by Heterogeneous Workers

In the previous section, we considered the setting of homogeneous workers, where the comparison results are determined only by the intrinsic latent scores of items but not by the characteristics of workers. However, on crowdsourcing platforms, the quality of the workers varies a lot. Some workers are less reliable or lack of the domain knowledge; some workers are spammers, who either do not actually take a look at the assigned pairs or are robots pretending to be human workers, and thus provide random comparison results in order to quickly receive payment; some workers may be poorly informed (or even malicious), misunderstand the ranking criteria and thus always flip the comparison results. To identify the reliability of a worker, one can assign the same pair of items to multiple workers and hope to identify the unreliable ones whose labels are often different from the majority. However, the abuse of this strategy will result in hiring too many workers and lead to a quick growth of the monetary cost. In order to maximize the accuracy of the final ranking under the limited amount of budget, it is critical to balance the budget spent on estimating the reliability of the workers and learning the true ranking of the items. To formalize such trade-off, we incorporate the reliability of each worker to our previous Bayesian MDP and generalize the AKG policy to the heterogeneity of workers.

2.4.1 Model Setup

Similar to the previous setting, we assume that each item i has an unknown latent score $\theta_i > 0$ for $i = 1, 2, \ldots, K$ which determines its true ranking π^* (see (2.1)) and $\boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha}^0)$. In the setting of heterogeneous workers, we assume that there are M crowd workers in total, denoted by $w = 1, 2, \ldots, M$. If a pair of items i and j with $i < j$ is presented to the worker w, we denote the returned comparison result by a random variable Y_{ij}^w such that

$$Y_{ij}^w = \begin{cases} 1 & \text{if item } i \text{ is preferred to item } j \text{ by worker } w \\ -1 & \text{if item } j \text{ is preferred to item } i \text{ by worker } w. \end{cases} \tag{2.33}$$

To model the reliability for workers, we introduce M latent parameters $\boldsymbol{\rho} = (\rho_1, \rho_2, \ldots, \rho_M)^T$ of reliability with $\rho_w \in [0, 1]$ for worker w and assume Y_{ij}^w has the following distribution

$$\Pr(Y_{ij}^w = 1) \quad = \quad \rho_w \frac{\theta_i}{\theta_i + \theta_j} + (1 - \rho_w) \frac{\theta_j}{\theta_i + \theta_j} \tag{2.34}$$

$$\Pr(Y_{ij}^w = -1) \quad = \quad \rho_w \frac{\theta_j}{\theta_i + \theta_j} + (1 - \rho_w) \frac{\theta_i}{\theta_i + \theta_j} \tag{2.35}$$

for $1 \leq i < j \leq K$ and $w = 1, 2, \ldots, M$. This model can be viewed as a combination of Dawid-Skene model for categorical labeling tasks [32, 95, 63] and Bradley-Terry-Luce (BTL) model, which was first introduced in [20]. Such a mixture of BTL model is flexible and capable of modeling various types of workers. When $\rho_w = 1$, the distribution in (2.34) and (2.35) reduces to (2.3), and we refer to worker w with $\rho_w = 1$ as a "fully reliable" worker[2]. Therefore, the reliability parameter ρ_w can be interpreted as the probability that worker w behaves as a random fully reliable workers in the previous section, namely, the one whose preference over a pair i and j follows a distribution in accordance with the BTL model (2.3). The worker with ρ_w closer to 1 is considered to be more reliable while a worker with ρ_w closer to 0 tends to be a poorly informed (or malicious) one who intentionally gives answers oppositive to the majority (truth). Also, a worker is known as a spammer if the associated ρ_w is near 0.5 since this worker prefers i or j in any pair i and j with an equal probability regardless of their latent scores.

The reliability of each worker is unknown for the ranking task, which needs to be gradually identified during the comparison process. In the Bayesian framework, since the reliability parameter ρ_w is supported on $[0, 1]$, it can be naturally modeled to follow a Beta prior distribution, i.e., $\rho_w \sim \mathcal{B}(\mu_w^0, \nu_w^0)$, for $w = 1, 2, \ldots, M$, where $\boldsymbol{\mu}^0 = (\mu_1^0, \mu_2^0, \ldots, \mu_M^0)$ and $\boldsymbol{\nu}^0 = (\nu_1^0, \nu_2^0, \ldots, \nu_M^0)$ are positive parameters.

2.4.2 Bayesian Decision Process

In this section, we model the sequential decision problem with a finite budget of T in the setting of heterogeneous workers. Since the workers now have different levels of reliability, we can no longer randomly select a worker from the crowd in each stage. Instead, we need to adaptively determine not only which pair of items to be compared but also who should perform this comparison task according to the historical results so that the budget can be gradually shifted towards more reliable workers.

Suppose a pair of items (i_t, j_t) with $i_t < j_t$ is compared by a worker w_t in stage t and the comparison result is $Y_{i_t j_t}^{w_t}$ defined in (2.33). The historical comparison results up to stage t can be summarized by a $K \times K \times M$ tensor $\mathbf{M}^t$, which is updated iteratively as follows. In particular, at each stage t, we define $\boldsymbol{\Delta}^t$ to be a sparse $K \times K \times M$ tensor with only non-zero element: if $Y_{i_t j_t}^{w_t} = 1$, $\boldsymbol{\Delta}_{i_t j_t w_t}^t = 1$ and if $Y_{i_t j_t}^{w_t} = -1$, $\boldsymbol{\Delta}_{j_t i_t w_t}^t = 1$. Let

$$\mathbf{M}^0 = \mathbf{0}, \quad \mathbf{M}^{t+1} = \mathbf{M}^t + \boldsymbol{\Delta}^t \quad \text{for } t = 0, 1, \ldots, T-1, \tag{2.36}$$

where $\mathbf{0}$ is a $K \times K \times M$ all-zero tensor. In contrast to the matrix M^t in (2.4), each element in the tensor $\mathbf{M}^t$ takes the value either zero or one because each worker is not allowed to compare the same pair more than once. The dynamic budget allocation policy is denoted by $\mathcal{A} = \{(i_t, j_t, w_t)\}_{t=0,1,\ldots,T-1}$ where $(i_t, j_t, w_t) = (i_t(\mathbf{M}^t), j_t(\mathbf{M}^t), w_t(\mathbf{M}^t))$ depends on the previous comparison results through $\mathbf{M}^t$. The

[2] We note that the full reliability does not imply that the worker is capable of identifying the latent scores of items and always give the correct comparison result, i.e., preferring the item with a higher latent score. Instead, being fully reliable only means the worker tries her best to provide the preference after a careful consideration, and the inconsistency of comparisons among workers is mainly because the intrinsic ambiguity of the pair of items.

posterior distributions of $\boldsymbol{\theta}$ and $\boldsymbol{\rho}$ in stage t are denoted by $p(\boldsymbol{\theta}|\mathbf{M}^t,\boldsymbol{\alpha}^0,\boldsymbol{\mu}^0,\boldsymbol{\nu}^0)$ and $p(\boldsymbol{\rho}|\mathbf{M}^t,\boldsymbol{\alpha}^0,\boldsymbol{\mu}^0,\boldsymbol{\nu}^0)$, respectively.

Similar to the homogeneous worker setup, we adopt the Kendall's tau (2.5) to measure the ranking accuracy. At each stage t, we denote the maximum posterior expected ranking accuracy by (with a slight abuse of notation)

$$\begin{aligned} h(\mathbf{M}^t) &\equiv \max_{\pi}\mathbb{E}\ \ \tau(\pi,\pi^*)|\mathbf{M}^t,\boldsymbol{\alpha}^0,\boldsymbol{\mu}^0,\boldsymbol{\nu}^0 \\ &= \max_{\pi}\frac{2\quad {}_{i\neq j}\,\mathbf{1}_{\pi(i)>\pi(j)}\Pr\left(\theta_i>\theta_j|\mathbf{M}^t,\boldsymbol{\alpha}^0,\boldsymbol{\mu}^0,\boldsymbol{\nu}^0\right)}{K(K-1)}. \end{aligned} \tag{2.37}$$

The maximizer in (2.37) is the optimal ranking inferred from the historical comparison results up to the stage t. Our goal is to search for the optimal policy $\mathcal{A}$ that maximizes the final expected ranking accuracy $h(\mathbf{M}^T)$, i.e.,

$$\max_{\mathcal{A}}\mathbb{E}^{\mathcal{A}}\ \ h(\mathbf{M}^T)|\boldsymbol{\alpha}^0,\boldsymbol{\mu}^0,\boldsymbol{\nu}^0\ \ . \tag{2.38}$$

This maximization problem can be further reformulated in a telescopic sum

$$h(\mathbf{M}^0)+\max_{\mathcal{A}}\mathbb{E}\ \mathop{\quad}\limits_{t=0}^{T-1}\ \mathbb{E}\ \ R(\mathbf{M}^t,i_t,j_t,w_t,Y_{i_tj_t}^{w_t})\ \mathbf{M}^t,\boldsymbol{\alpha}^0,\boldsymbol{\mu}^0,\boldsymbol{\nu}^0\quad \boldsymbol{\alpha}^0,\boldsymbol{\mu}^0,\boldsymbol{\nu}^0\ \ , \tag{2.39}$$

where

$$R(\mathbf{M}^t,i_t,j_t,w_t,Y_{i_tj_t}^{w_t})\equiv h(\mathbf{M}^{t+1})-h(\mathbf{M}^t), \tag{2.40}$$

is the *stage-wise reward* depending on $\mathbf{M}^t$,i_t,j_t,w_t and $Y_{i_tj_t}^{w_t}$. It can be interpreted as the improvement of the expected ranking accuracy after receiving the comparison result at stage t. The *state variable* of the MDP (2.38) or (2.39) is the tensor $\mathbf{M}^t$ which evolves according to (2.36) and the state space at each t is

$$\mathcal{S}^t=\ \ \mathbf{M}\in\{0,1\}^{K\times K\times M}:\ \mathop{\quad}\limits_{i,j,w}\mathbf{M}_{ijw}=t\ \ .$$

The *expected transition probabilities* of MDP (2.38) are

$$\mathbb{E}\ \ \Pr(Y_{ij}^w=1)|\mathbf{M}^t,\boldsymbol{\alpha}^0,\boldsymbol{\mu}^0,\boldsymbol{\nu}^0\ \ =\mathbb{E}\ \ \rho_w\frac{\theta_i}{\theta_i+\theta_j}+(1-\rho_w)\frac{\theta_j}{\theta_i+\theta_j}\ \mathbf{M}^t,\boldsymbol{\alpha}^0,\boldsymbol{\mu}^0,\boldsymbol{\nu}^0 \tag{2.41}$$

$$\mathbb{E}\ \ \Pr(Y_{ij}^w=-1)|\mathbf{M}^t,\boldsymbol{\alpha}^0,\boldsymbol{\mu}^0,\boldsymbol{\nu}^0\ \ =\mathbb{E}\ \ \rho_w\frac{\theta_j}{\theta_i+\theta_j}+(1-\rho_w)\frac{\theta_i}{\theta_i+\theta_j}\ \mathbf{M}^t,\boldsymbol{\alpha}^0,\boldsymbol{\mu}^0,\boldsymbol{\nu}^0 \tag{2.42}$$

for $i,j=1,2,\ldots,K$ and $w=1,2,\ldots,M$. So far, we have modeled the sequential budget allocation in the heterogeneous worker setting as a Bayesian MDP. Due to the similar reasons that have been explained in Section 2.3.2, although the dynamic programming can be directly applied to solve the Bayesian MDP and obtain the optimal policy, it is computationally intractable. In fact, the Bayesian MDP (2.39) is even more challenging to solve than that for the homogeneous worker setting due to a much larger state space after introducing the reliability of workers. In the next subsection, we will propose a computationally efficient approximated knowledge gradient policy for (2.39).

2.4.3 Approximated Knowledge Gradient Policy

To solve the Bayesian MDP (2.39), we still consider the family of knowledge gradient (KG) policies. In our problem, the KG policy will select the pair of items and the worker that together give the highest expected stage-wise reward. In particular, at the t-stage, the KG policy for (2.39) will choose the pair (i_t, j_t) and the worker w_t such that

$$\begin{aligned}(i_t, j_t, w_t) \quad &\in \quad \operatorname*{arg\,max}_{i<j,w} \mathbb{E} \;\; R(\mathbf{M}^t, i, j, w, Y_{ij}^w) \; \mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0 \qquad (2.43)\\ &= \quad \operatorname*{arg\,max}_{i<j,w} \Big\{ \mathbb{E} \;\; \Pr(Y_{ij}^w = 1)|\mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0 \;\; R(\mathbf{M}^t, i, j, w, 1)\\ &\qquad\qquad + \mathbb{E} \;\; \Pr(Y_{ij}^w = -1)|\mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0 \;\; R(\mathbf{M}^t, i, j, w, -1) \Big\}.\end{aligned}$$

To implement the KG policy (2.43), we encounter the same difficulties as when we implemented (2.16). Specifically, since the posterior distributions $p(\boldsymbol{\theta}|\mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0)$ and $p(\boldsymbol{\rho}|\mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0)$ are sophisticated and the MAX-LOP problem (2.37) is NP-hard, we cannot efficiently evaluate the stage-wise reward (2.40) and the transition probabilities (2.41) and (2.42). To obtain a computationally efficient policy, we follow the techniques in Section 2.3.3 to approximate the posterior distributions $p(\boldsymbol{\theta}|\mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0)$ and $p(\rho_w|\mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0)$ recursively using a sequence of Dirichlet distributions $\mathrm{Dir}(\boldsymbol{\alpha}^t)$ and a sequence of beta distributions $\mathcal{B}(\mu_w^t, \nu_w^t)$, respectively, for $w = 1, 2, \ldots, M$ and $t = 1, 2, \ldots, T$. The parameters $\boldsymbol{\alpha}^t(\alpha_1^t, \alpha_2^t, \ldots, \alpha_K^t)$, $\boldsymbol{\mu}^t = (\mu_1^t, \mu_2^t, \ldots, \mu_M^t)$ and $\boldsymbol{\nu}^t = (\nu_1^t, \nu_2^t, \ldots, \nu_M^t)$ will be chosen recursively based on moment matching.

Suppose $\boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha})$ for some parameter vector $\boldsymbol{\alpha} \in \mathbb{R}^K$ and $\rho_w \sim \mathcal{B}(\mu_w, \nu_w)$ for each w with $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_M)$ and $\boldsymbol{\nu} = (\nu_1, \nu_2, \ldots, \nu_M)$. We consider a basic scenario where only one comparison result Y_{ij}^w from worker w for a pair (i, j) has been observed. We can approximate $p(\boldsymbol{\theta}|Y_{ij}^w, \boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\nu})$ by a Dirichlet distribution $\mathrm{Dir}(\boldsymbol{\alpha}')$ and $p(\rho_w|Y_{ij}^w, \boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\nu})$ by a Beta distribution $\mathcal{B}(\mu_w', \nu_w')$ such that

$$\begin{aligned}\mathbb{E}\left[\theta_k|\boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha}')\right] \quad &= \quad \mathbb{E}[\theta_k|Y_{ij}^w, \boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\nu}] \text{ for } k = 1, 2, \ldots, K \qquad (2.44)\\ \mathbb{E} \quad \underset{k=1}{\overset{K}{}} \;\; \theta_k^2|\boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha}') \quad &= \quad \mathbb{E} \quad \underset{k=1}{\overset{K}{}} \;\; \theta_k^2|Y_{ij}^w, \boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\nu} \qquad (2.45)\\ \mathbb{E}\left[\rho_w|\rho_w \sim \mathcal{B}(\mu_w', \nu_w')\right] \quad &= \quad \mathbb{E} \;\; \rho_w|Y_{ij}^w, \boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\nu} \qquad (2.46)\\ \mathbb{E} \;\; \rho_w^2 + (1-\rho_w)^2|\rho_w \sim \mathcal{B}(\mu_w', \nu_w') \quad &= \quad \mathbb{E} \;\; \rho_w^2 + (1-\rho_w)^2|Y_{ij}^w, \boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\nu} \;\; . \qquad (2.47)\end{aligned}$$

Note that we do not need to approximate $p(\rho_w \;|Y_{ij}^w, \boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\nu})$ for $w' = w$ since the worker w' has not performed any comparison so that $p(\rho_w \;|Y_{ij}^w, \boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\nu})$ is still the prior distribution $\mathcal{B}(\mu_w \;, \nu_w \;)$. This system of equations has the following explicit characterization.

Proposition 3. *Suppose* $\boldsymbol{\theta} \sim$ *Dir*$(\boldsymbol{\alpha})$ *and* $\rho_w \sim \mathcal{B}(\mu_w, \nu_w)$ *for worker* w *and* Y_{ij}^w *is the only comparison result. Let* $\alpha_0 = \;{}_{k=1}^{K} \alpha_k$ *and* $\alpha_0' = \;{}_{k=1}^{K} \alpha_k'$. *The equations* (2.44), (2.45), (2.46) *and* (2.47) *can be* *represented as*

$$\left\{\begin{array}{rcl}
\frac{\alpha_i}{\alpha_0} & = & \eta_{ijw}\frac{(\alpha_i+1)(\alpha_i+\alpha_j)}{\alpha_0(\alpha_i+\alpha_j+1)}+(1-\eta_{ijw})\frac{\alpha_i(\alpha_i+\alpha_j)}{\alpha_0(\alpha_i+\alpha_j+1)}\\
\frac{\alpha_j}{\alpha_0} & = & \eta_{ijw}\frac{\alpha_j(\alpha_i+\alpha_j)}{\alpha_0(\alpha_i+\alpha_j+1)}+(1-\eta_{ijw})\frac{(\alpha_j+1)(\alpha_i+\alpha_j)}{\alpha_0(\alpha_i+\alpha_j+1)}\\
\frac{\alpha_k}{\alpha_0} & = & \frac{\alpha_k}{\alpha_0} \quad \textit{for } k=i,j\\
{}_{k=1}^{K}\frac{\alpha_k(\alpha_k+1)}{\alpha_0(\alpha_0+1)} & = & \eta_{ijw}\frac{(\alpha_i+1)(\alpha_i+2)(\alpha_i+\alpha_j)}{\alpha_0(\alpha_0+1)(\alpha_i+\alpha_j+2)}+(1-\eta_{ijw})\frac{\alpha_i(\alpha_i+1)(\alpha_i+\alpha_j)}{\alpha_0(\alpha_0+1)(\alpha_i+\alpha_j+2)}\\
 & & +\eta_{ijw}\frac{\alpha_j(\alpha_j+1)(\alpha_i+\alpha_j)}{\alpha_0(\alpha_0+1)(\alpha_i+\alpha_j+2)}+(1-\eta_{ijw})\frac{(\alpha_j+1)(\alpha_j+2)(\alpha_i+\alpha_j)}{\alpha_0(\alpha_0+1)(\alpha_i+\alpha_j+2)}\\
 & & + \quad {}_{k\neq i,j}\frac{\alpha_k(\alpha_k+1)}{\alpha_0(\alpha_0+1)}\\
\frac{\mu_w}{\mu_w+\nu_w} & = & \eta_{ijw}\frac{\mu_w+(1+Y_{ij}^w)/2}{\mu_w+\nu_w+1}+(1-\eta_{ijw})\frac{\mu_w+(1-Y_{ij}^w)/2}{\mu_w+\nu_w+1}\\
\frac{\mu_w(\mu_w+1)+\nu_w(\nu_w+1)}{(\mu_w+\nu_w)(\mu_w+\nu_w+1)} & = & \eta_{ijw}\frac{(\mu_w+(1+Y_{ij}^w)/2)(\mu_w+(3+Y_{ij}^w)/2)}{(\mu_w+\nu_w+1)(\mu_w+\nu_w+2)}\\
 & & +(1-\eta_{ijw})\frac{(\mu_w+(1-Y_{ij}^w)/2)(\mu_w+(3-Y_{ij}^w)/2)}{(\mu_w+\nu_w+1)(\mu_w+\nu_w+2)}\\
 & & +\eta_{ijw}\frac{(\nu_w+(1-Y_{ij}^w)/2)(\nu_w+(3-Y_{ij}^w)/2)}{(\mu_w+\nu_w+1)(\mu_w+\nu_w+2)}\\
 & & +(1-\eta_{ijw})\frac{(\nu_w+(1+Y_{ij}^w)/2)(\nu_w+(3+Y_{ij}^w)/2)}{(\mu_w+\nu_w+1)(\mu_w+\nu_w+2)}.
\end{array}\right. \tag{2.48}$$

where $\eta_{ijw}=\frac{[(1+Y_{ij}^w)\mu_w+(1-Y_{ij}^w)\nu_w]\alpha_i}{[(1+Y_{ij}^w)\mu_w+(1-Y_{ij}^w)\nu_w]\alpha_i+[(1+Y_{ij}^w)\nu_w+(1-Y_{ij}^w)\mu_w]\alpha_j}$.

The proof of Proposition 3 is given in Appendix. We denote any $\boldsymbol{\alpha}'$, μ'_w and ν'_w that satisfy (2.44), (2.45), (2.46) and (2.47), and thus (2.48), by

$$\boldsymbol{\alpha}'=\mathbf{MM}_\alpha(\boldsymbol{\alpha},i,j,w,Y_{ij}^w) \quad\text{and}\quad (\mu'_w,\nu'_w)=\mathbf{MM}_{\mu\nu}(\boldsymbol{\alpha},i,j,w,Y_{ij}^w). \tag{2.49}$$

Although the equations in Proposition 3 are more complicated than those in Proposition 1, the right-hand sides of (2.48) are still constants for any given i, j, w, Y_{ij}^w, $\boldsymbol{\alpha}$, μ_w and ν_w so that both $\boldsymbol{\alpha}'=\mathbf{MM}_\alpha(\boldsymbol{\alpha},i,j,w,Y_{ij}^w)$ and $(\mu'_w,\nu'_w)=\mathbf{MM}_{\mu\nu}(\boldsymbol{\alpha},i,j,w,Y_{ij}^w)$ can be solved in a closed form. In fact, we denote the constants on the right-hand sides of (2.20) as C_i, C_j, C_k (for $k=i,j$), D, E and F, respectively. It is easy to see that ${}_{k=1}^{K}C_k=1$. By the same derivation for (2.22), we obtain the following closed form for $\boldsymbol{\alpha}'=\mathbf{MM}_\alpha(\boldsymbol{\alpha},i,j,w,Y_{ij}^w)$

$$\alpha'_0=\frac{D-1}{{}_{k=1}^{K}C_k^2-D} \quad\text{and}\quad \alpha'_k=C_k\alpha'_0 \text{ for } k=1,2,\ldots,K, \tag{2.50}$$

which takes the same form as (2.22) but with the constants C_k for $k=1,2,\ldots,K$ defined differently (which involve the information of worker w, i.e., μ_w and ν_w). Similarly, solving μ'_w and ν'_w from the last two equations in (2.48), we obtain the following closed form for $(\mu'_w,\nu'_w)=\mathbf{MM}_{\mu\nu}(\boldsymbol{\alpha},i,j,w,Y_{ij}^w)$

$$\mu'_w=\frac{(F-1)E}{E^2+(1-E)^2-F} \quad\text{and}\quad \nu'_w=\frac{(F-1)(1-E)}{E^2+(1-E)^2-F}. \tag{2.51}$$

Although the approximate scheme above is derived when there is only one comparison result, it generates a Dirichlet distribution $\text{Dir}(\boldsymbol{\alpha}')$ for $\boldsymbol{\theta}$ and a Beta distribution $\mathcal{B}(\mu'_w,\nu'_w)$ for ρ_w and does not change the Beta distribution $\mathcal{B}(\mu_w\ ,\nu_w\ )$ for $w'=w$. The fact that the approximated posteriors take the same form as the priors suggests that we can apply this approximation scheme iteratively to approximate $p(\boldsymbol{\theta}|\mathbf{M}^t,\boldsymbol{\alpha}^0,\boldsymbol{\mu}^0,\boldsymbol{\nu}^0)$

and $p(\boldsymbol{\rho}|\mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0)$ for any given policy $\mathcal{A} = \{(i_t, j_t, w_t)\}_{t=0,1,\ldots,T-1}$. In particular, let $\boldsymbol{\alpha}^t$, $\boldsymbol{\mu}^t$ and $\boldsymbol{\nu}^t$ be the sequences of parameters generated recursively as follows

$$\begin{aligned} \boldsymbol{\alpha}^{t+1} &= \mathbf{MM}_\alpha(\boldsymbol{\alpha}^t, i_t, j_t, w_t, Y_{i_t j_t}^{w_t}) && (2.52) \\ (\mu_w^{t+1}, \nu_w^{t+1}) &= \begin{cases} \mathbf{MM}_{\mu\nu}(\boldsymbol{\alpha}^t, i_t, j_t, w_t, Y_{i_t j_t}^{w_t}) & \text{if } w = w_t \\ (\mu_w^t, \nu_w^t) & \text{if } w = w_t \end{cases} && (2.53) \end{aligned}$$

for $t = 1, 2, \ldots, T$. The posterior distributions $p(\boldsymbol{\theta}|\mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0)$ and $p(\boldsymbol{\rho}|\mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0)$ can be approximated by $\mathrm{Dir}(\boldsymbol{\alpha}^t)$ and $\Pi_{w=1,\ldots,M}\mathcal{B}(\mu_w^t, \nu_w^t)$, respectively.

Following the same strategy as in (2.24) and (2.25), we can approximate (2.41) and (2.42) as

$$\begin{aligned} & \mathbb{E}\ \ \Pr(Y_{ij}^w = 1)\ \mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0 && (2.54) \\ \approx\ & \mathbb{E}\ \ \rho_w \frac{\theta_i}{\theta_i + \theta_j} + (1 - \rho_w)\frac{\theta_j}{\theta_i + \theta_j}\ \ \boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha}^t), \rho_w \sim \mathcal{B}(\mu_w^t, \nu_w^t) \\ =\ & \frac{\mu_w^t}{\mu_w^t + \nu_w^t}\frac{\alpha_i^t}{\alpha_i^t + \alpha_j^t} + \frac{\nu_w^t}{\mu_w^t + \nu_w^t}\frac{\alpha_j^t}{\alpha_i^t + \alpha_j^t} \end{aligned}$$

and

$$\begin{aligned} & \mathbb{E}\ \ \Pr(Y_{ij}^w = -1)\ \mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0 && (2.55) \\ \approx\ & \mathbb{E}\ \ \rho_w \frac{\theta_j}{\theta_i + \theta_j} + (1 - \rho_w)\frac{\theta_i}{\theta_i + \theta_j}\ \ \boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha}^t), \rho_w \sim \mathcal{B}(\mu_w^t, \nu_w^t) \\ =\ & \frac{\mu_w^t}{\mu_w^t + \nu_w^t}\frac{\alpha_j^t}{\alpha_i^t + \alpha_j^t} + \frac{\nu_w^t}{\mu_w^t + \nu_w^t}\frac{\alpha_i^t}{\alpha_i^t + \alpha_j^t} \end{aligned}$$

and approximate $\Pr\left(\theta_i > \theta_j|\mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0\right)$ in (2.37) as

$$\Pr\left(\theta_i > \theta_j|\mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0\right) \approx \Pr\left(\theta_i > \theta_j|\boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha}^t)\right) = I_{\frac{1}{2}}(\alpha_j^t, \alpha_i^t). \tag{2.56}$$

The approximation (2.56) helps to simplify the NP-hard MAX-LOP in (2.37) as

$$\max_\pi \mathbb{E}\ \ \tau(\pi, \pi^*)|\mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0\ \ \approx \max_\pi \mathbb{E}\ \ \tau(\pi, \pi^*)|\boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha}^t)\ \ ,$$

where the right-hand side can be solved easily by sorting of the components of $\boldsymbol{\alpha}^t$ according to Theorem 2.

Similar to (2.29), the stage-wise reward is approximated as

$$\begin{aligned} R(\mathbf{M}^t, i, j, w, Y_{ij}^w) &= \max_\pi \mathbb{E}\ \ \tau(\pi, \pi^*)|\mathbf{M}^{t+1}, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0\ \ - \max_\pi \mathbb{E}\ \ \tau(\pi, \pi^*)|\mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0 \\ &\approx \max_\pi \mathbb{E}\left[\tau(\pi, \pi^*)|\boldsymbol{\theta} \sim \mathrm{Dir}(\hat{\boldsymbol{\alpha}})\right] - \max_\pi \mathbb{E}\ \ \tau(\pi, \pi^*)|\boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha}^t) \\ &= \frac{2}{K(K-1)}\left(\ \ _{\pi_{\hat{\boldsymbol{\alpha}}}(i\ )>\pi_{\hat{\boldsymbol{\alpha}}}(j\ )}\ \ I_{\frac{1}{2}}(\hat{\alpha}_j\ , \hat{\alpha}_i\ ) - \ \ _{\pi_{\boldsymbol{\alpha}^t}(i\ )>\pi_{\boldsymbol{\alpha}^t}(j\ )}\ \ I_{\frac{1}{2}}(\alpha_j^t\ , \alpha_i^t\ )\right) \\ &\equiv \tilde{R}(\boldsymbol{\alpha}^t, i, j, w, Y_{ij}^w) \end{aligned} \tag{2.57}$$

where $\hat{\boldsymbol{\alpha}} = \mathbf{MM}_\alpha(\boldsymbol{\alpha}^t, i, j, w, Y_{ij}^w)$. Putting (2.54), (2.55), (2.56) and (2.57) together, we can approximate

Algorithm 2 Approximated Knowledge Gradient Policy with Heterogeneous Workers

```
Initialization: Choose $\boldsymbol{\alpha}^0$, $\boldsymbol{\mu}^0$ and $\boldsymbol{\nu}^0$ for the prior distributions.
For $t = 0, \ldots, T-1$ do
    For each pair $(i, j)$ with $i < j$, compute $\tilde{R}(\boldsymbol{\alpha}^t, i, j, w, 1)$ and $\tilde{R}(\boldsymbol{\alpha}^t, i, j, w, -1)$ according to (2.57).
    Select $(i_t, j_t, w_t)$ such that
        $(i_t, j_t) \in \arg\max_{i<j,\, w} \Big[ \Big( \frac{\mu_w^t}{\mu_w^t + \nu_w^t} \frac{\alpha_i^t}{\alpha_i^t + \alpha_j^t} + \frac{\nu_w^t}{\mu_w^t + \nu_w^t} \frac{\alpha_j^t}{\alpha_i^t + \alpha_j^t} \Big) \tilde{R}(\boldsymbol{\alpha}^t, i, j, w, 1)$
        $+ \Big( \frac{\mu_w^t}{\mu_w^t + \nu_w^t} \frac{\alpha_j^t}{\alpha_i^t + \alpha_j^t} + \frac{\nu_w^t}{\mu_w^t + \nu_w^t} \frac{\alpha_i^t}{\alpha_i^t + \alpha_j^t} \Big) \tilde{R}(\boldsymbol{\alpha}^t, i, j, w, -1) \Big]$    (2.59)
    and present item $i_t$ and item $j_t$ to worker $w_t$ and receive the comparison result $Y_{i_t j_t}^{w_t}$.
    According to (2.49), (2.50) and (2.51), compute
        $\boldsymbol{\alpha}^{t+1} = \mathbf{MM}_\alpha(\boldsymbol{\alpha}^t, i_t, j_t, w_t, Y_{i_t j_t}^{w_t})$    (2.60)
        $(\mu_w^{t+1}, \nu_w^{t+1}) = \begin{cases} \mathbf{MM}_{\mu\nu}(\boldsymbol{\alpha}^t, i_t, j_t, w_t, Y_{i_t j_t}^{w_t}) & \text{if } w = w_t \\ (\mu_w^t, \nu_w^t) & \text{if } w \neq w_t \end{cases}$    (2.61)
End For
Return: The aggregated ranking $\pi_{\boldsymbol{\alpha}^T}$ obtained by sorting the components of $\boldsymbol{\alpha}^T$.
```

the expected stage-wise reward $\mathbb{E} \;\; R(\mathbf{M}^t, i, j, w, Y_{ij}^w) \;\; M^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0$ as

$$
\begin{aligned}
& \mathbb{E} \;\; R(\mathbf{M}^t, i, j, w, Y_{ij}^w) \;\; M^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0 \\
= \;\; & \mathbb{E} \;\; \Pr(Y_{ij}^w = 1) \;\; \mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0 \;\; R(\boldsymbol{\alpha}^t, i, j, w, 1) \\
& + \mathbb{E} \;\; \Pr(Y_{ij}^w = -1) \;\; \mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0 \;\; R(\boldsymbol{\alpha}^t, i, j, w, -1) \\
\approx \;\; & \frac{\mu_w^t}{\mu_w^t + \nu_w^t} \frac{\alpha_i^t}{\alpha_i^t + \alpha_j^t} + \frac{\nu_w^t}{\mu_w^t + \nu_w^t} \frac{\alpha_j^t}{\alpha_i^t + \alpha_j^t} \quad \tilde{R}(\boldsymbol{\alpha}^t, i, j, w, 1) \\
+ \;\; & \frac{\mu_w^t}{\mu_w^t + \nu_w^t} \frac{\alpha_j^t}{\alpha_i^t + \alpha_j^t} + \frac{\nu_w^t}{\mu_w^t + \nu_w^t} \frac{\alpha_i^t}{\alpha_i^t + \alpha_j^t} \quad \tilde{R}(\boldsymbol{\alpha}^t, i, j, w, -1). \qquad (2.58)
\end{aligned}
$$

When the workers have various levels of reliability, our AKG policy will choose the pair (i_t, j_t) and present it to worker w_t so that (2.58) is maximized. The AKG policy for the setting of heterogeneous workers is formally presented as Algorithm 3. Note that when $\rho_w = 1$ for all w, we do not need to solve (2.46) and (2.47) anymore and thus the rest of the problem reduces to the homogeneous setting.

2.5 Experiment

In this section, we conduct empirical studies using both simulated and real data. We compare the proposed AKG algorithms to some existing methods in terms of ranking accuracy versus different levels of budget as well as computation time. We also show some interesting properties of the proposed AKG policies, e.g., how budget will be allocated over pairs of items with different levels of ambiguity and workers with different levels of reliability. The ranking accuracy is evaluated using the Kendall's tau as defined in (2.5).

2.5.1 Simulated Study under the Homogeneous Workers Setting

In this section, we assume that all workers are fully reliable and investigate the performance of the AKG policy (Algorithm 1). Two scenarios are designed: 10 items with a total budget of 100, and 100 items with a total budget of 1000. Each scenario consists of 100 independent trials and the average ranking accuracy is reported. For each trial, the latent item scores $\boldsymbol{\theta}$ is sampled uniformly from the simplex in (2.6), which determines the true ranking π^*. Given $\boldsymbol{\theta}$, the comparison results are generated according to the Bradley-Terry-Luce model (2.3). We compare several different methods, including the proposed AKG, random sampling (uniformly random sampling), distance-based sampling, adaptive polling [88] and rank centrality with uniform sampling or knowledge gradient sampling [85]. The details of the methods are provided as follows.

1. **AKG** (see Algorithm 1): We set the prior of $\boldsymbol{\theta}$ to be the uniform distribution on the simplex (i.e., $\boldsymbol{\alpha}^0$ is set to be an all-one vector).

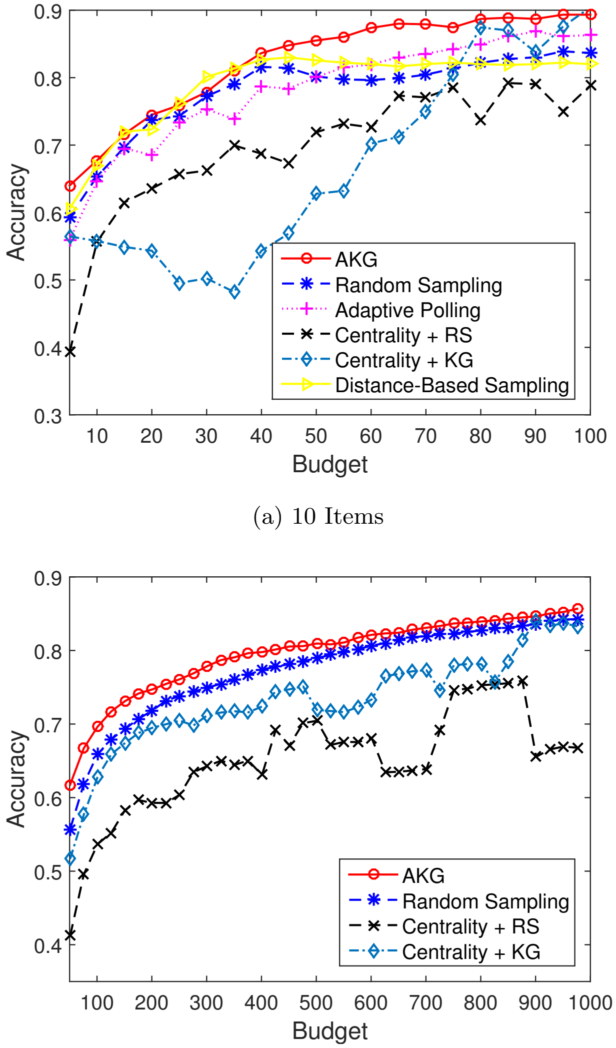
2. **Random Sampling**: The random sampling algorithm is similar to Algorithm 1 in terms of the posterior approximation (by moment matching) and rank inference (by sorting the approximated posterior parameters $\boldsymbol{\alpha}^t$) after receiving each label. The only difference is that this algorithm replaces Step 2 of Algorithm 1 by a random sampling policy, which selects (i_t, j_t) randomly at each stage. We also choose the uniform distribution on the simplex as the prior.

3. **Distance-Based Sampling**: This algorithm is also the same as Algorithm 1 in terms of the posterior approximation. However, in the sampling phase, this algorithm simply selects the pair of items (i_t, j_t) with the closest posterior parameters α_i^t and α_j^t. We choose the uniform distribution on the simplex as the prior.

4. **Adaptive Polling**: This is a greedy policy proposed by [88], which chooses the pair of items to maximize the KL-divergence between the posterior and prior. The initial $K \times K$ matrix M used in adaptive polling is set to 0 on the diagonal and 0.15 everywhere else.

5. **Rank Centrality**: This is a static rank aggregation algorithm recently proposed by [85]. We combine it with both the random sampling policy and the knowledge gradient policy. Specifically, for **Centrality + RS**, we randomly select a pair of items at each stage and infer the true ranking using rank centrality. For **Centrality + KG**, we select the next pair of items using AKG policy, but estimate the ranking using rank centrality.

It is worthwhile to point out that we are able to compute the optimal policy exactly only up to the 4-item case, which is not interesting from the ranking perspective and thus is left out from the experiment.



(a) 10 Items

(b) 100 Items

Figure 2.1: Performance comparison under the homogeneous workers setting. The x-axis is the budget level and y-axis is the averaged ranking accuracy.

As we can see from Figure 2.1, the AKG policy has higher accuracy than other methods at all budget levels. Note that the average accuracy of AKG surpasses the level of 70% with only 20 pairs in the case of 10 items. In general, random sampling has similar performance as AKG at the beginning, but eventually AKG will outperform random sampling as it will spend more budget on the ambiguous pairs. This will be verified in the next experiment. Meanwhile, if we combine rank centrality with knowledge gradient sampling, the performance of the algorithm can be boosted significantly. Furthermore, the curves of ranking accuracy of AKG are in general monotonically increasing and have fewer "bumps" than other algorithms. This implies that the sequence of posterior parameters $\boldsymbol{\alpha}^t$ is quite stable when the budget level becomes larger. We also note that due to the high computational cost of adaptive polling, it takes extremely long time when the

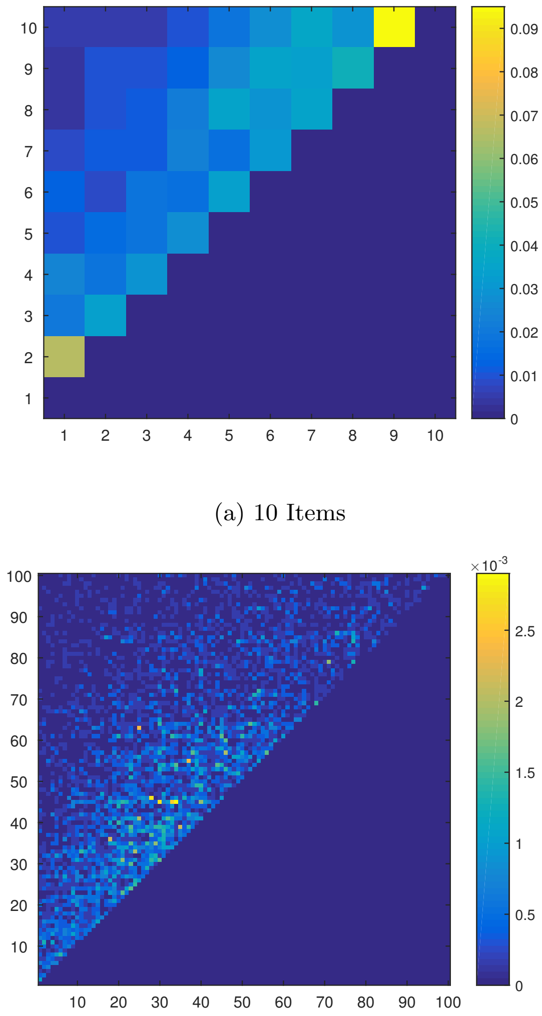
Table 2.1: Comparison in computation time under the homogeneous workers setting.

No. of Items	AKG	Adaptive Polling
10	0.023 sec	20 sec
25	0.75 sec	42 min
100	22 sec	-

number of items is 100 and thus we omit its performance in Figure 2.1b.

It is worthwhile to note that AKG runs significantly faster than the adaptive polling method. It enjoys the advantage of closed-form updating rule during each iteration/stage without using a numerical algorithm as a subroutine, which is a good feature for online applications. In contrast, adaptive polling is much slower because it requires inverting a $K \times K$ matrix for all $O(K^2)$ possible pairs and all possible comparison results in each iteration. Table 2.1 gives the computation time of a *single iteration* for both AKG and adaptive polling. Note that in the 25-item case, the computation time for adaptive polling of a single iteration has already exceeded 40 minutes. Therefore, we omit to present the computation time of adaptive polling when the number of items is 100 in Table 2.1 since each iteration/stage would take hours to run.

Next, we study the allocation of labeling budget over pairs of items with different levels of ambiguity when using the AKG policy. Again, we consider two scenarios: $K = 10, T = 100$ and $K = 100, T = 1000$, each with 100 independent trials. We report the averaged labeling frequency of each pair. The results are presented in Figure 2.2 in the form of heat maps. In Figure 2.2, each small block represents a pair of items. Items are sorted based on their true latent scores, from lowest to highest along both y-axis and x-axis, so that the item pairs along the back-diagonal are more ambiguous than those around the corner. Figure 2.2 presents the normalized number of comparisons over different pairs in total T stages. It can be seen from Figure 2.2 that the back-diagonal pairs in general have higher labeling frequency than other pairs. Some adjacent pairs are labeled 10 times more frequently than the distant pairs. To further demonstrate this property, we design a scenario in which out of 10 items, the two worst items and the two best items have very close true scores respectively. Although the main goal of the algorithm is to achieve higher ranking accuracy, we are still curious to see whether our policy can spend the budget on these two pairs. As we can see from Figure 2.3, it is clear that the algorithm concentrates on the 1-2 pair and the 9-10 pair. This implies that our policy can identify and explore more ambiguous pairs to improve the learning of the true ranks.



(a) 10 Items

(b) 100 Items

Figure 2.2: Heat map of labeling frequency for item pairs with different levels of ambiguity

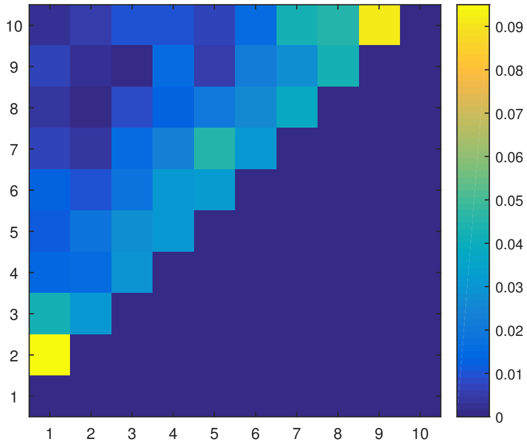
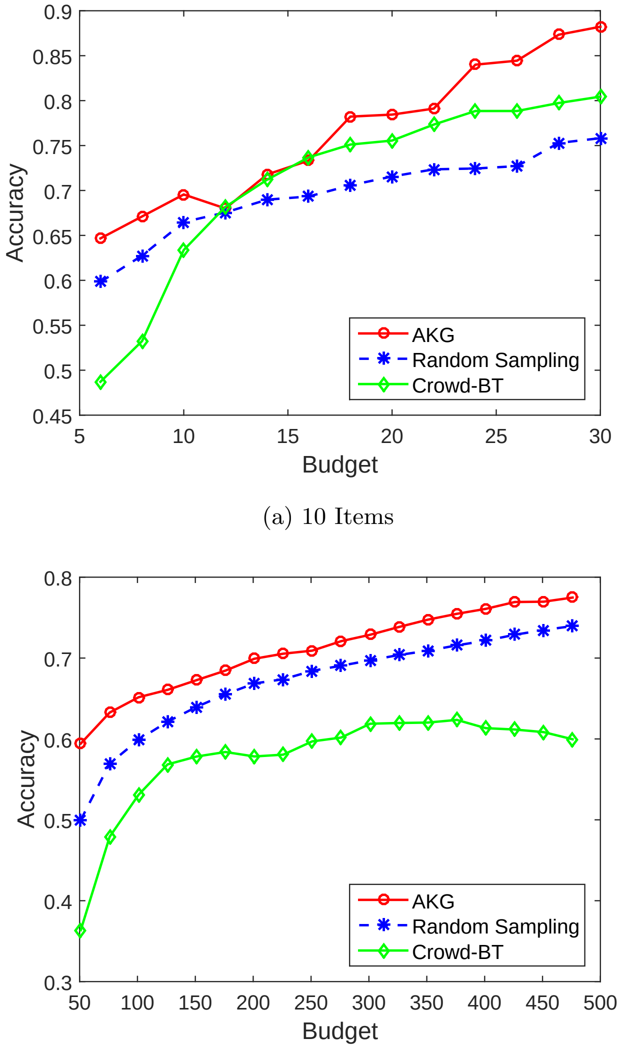


Figure 2.3: Heat map of labeling frequency for pairs with very close scores

2.5.2 Simulated Study under the Heterogeneous Workers Setting



(a) 10 Items

(b) 100 Items

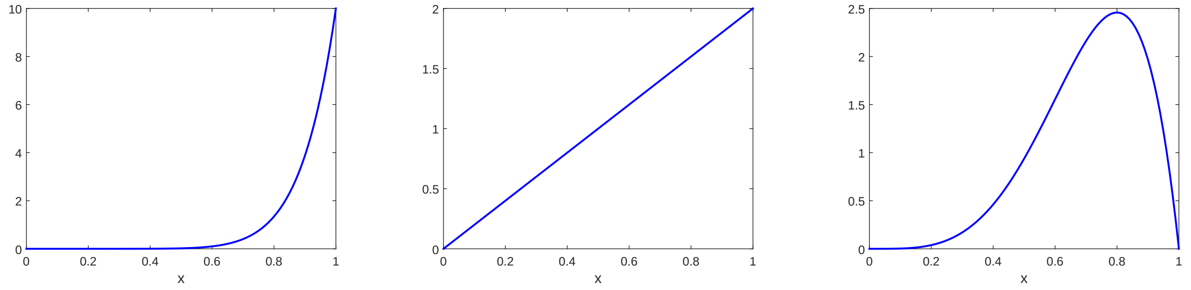
Figure 2.4: Performance comparison under the heterogeneous workers setting. The x-axis is the budget level

In this section we bring worker quality ρ_w into consideration, which is assumed to be drawn from the Beta(4,1) distribution. We choose the Beta(4,1) to generate ρ_w since the average reliability measure of workers in this case is $4/5 = 80\%$. This assumption is in line with the practice in that there are usually more reliable workers than unreliable ones. Similar to the homogeneous worker setting, we consider two scenarios: 10 items with 10 heterogeneous workers ($K = 10$, $M = 10$); 100 items with 50 heterogeneous workers ($K = 100$, $M = 50$) and we note that each worker is allowed to label any pair at most once. We compare the following three methods.

1. **AKG** (see Algorithm 3): We set the prior of $\boldsymbol{\theta}$ to be the uniform distribution on the simplex (i.e., $\boldsymbol{\alpha}^0$ is set to be an all-one vector) and choose $\mu_w^0 = 4, \nu_w^0 = 1$ for each worker $w = 1, 2, \ldots, M$.

2. **Random Sampling**: It is implemented simply by replacing Step 2 of Algorithm 3 by a random sampling policy, which selects a triplet $\{$item i, item j, worker $w\}$ uniformly randomly at each stage. The choices of priors are the same as in AKG. Like the AKG method, the random sampling algorithm also maintains a Dirichlet distribution for the scores of items and a beta distribution for the reliability parameter of each worker using moment matching.

3. **Crowd-BT**: This is an adaptive algorithm recently proposed by [20], which chooses the triplet $\{$item i, item j, worker $w\}$ at each iteration to maximize the information gain. This can be viewed as an extension of the adaptive polling [88] by incorporating the workers' reliability. Unlike adaptive polling which computes the relative entropy for each pair exactly, Crowd-BT uses moment matching to approximate the posterior and hence runs significantly faster than adaptive polling. The parameter γ, which balances the exploitation-exploration trade-off in [20], is set to 1 in this experiment.

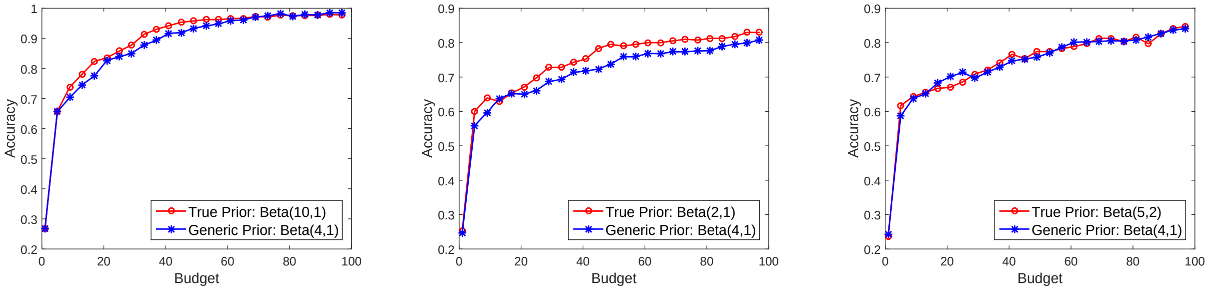


(a) Beta$(10, 1)$ (b) Beta$(2, 1)$ (c) Beta$(5, 2)$

Figure 2.5: Density plots of different Beta distributions for generating ρ_w

Table 2.2: Computation time under the heterogeneous workers setting.

No. of Items	No. of Workers	AKG
10	10	0.038 sec
25	20	0.82 sec
100	50	41 sec



(a) Beta(10, 1) Prior

(b) Beta(2, 1) Prior

(c) Beta(5, 2) Prior

Figure 2.6: Comparisons between AKG using Beta(4,1) prior and AKG using the true generating distribution as prior.

The comparison results are presented in Figure 2.4, where AKG outperforms the other two methods, especially when the budget level is low. The performance of random sampling is comparable to AKG at the beginning. As we gather more information, AKG can learn the reliability of workers so that the budget will be gradually shifted towards those reliable workers (as shown later in Figure 2.7). In fact, it can be seen from Figure 2.4 that the ranking accuracy of AKG increases more quickly than that of other methods. In this experiment, even if there is a small amount of budget (e.g. $T = K$), the AKG policy is still able to achieve reasonably good performance. We notice that in the 100-item case Crowd-BT is beaten by random sampling. The main reason is that when the reliability of workers varies and the pool is large, it is difficult to balance exploration and exploitation for Crowd-BT, which has already been acknowledged in [20]. Similar to the previous setting, we also give the table of the computation time of a *single iteration* for AKG in Table 2.2. As we can see from the table, even with another dimension of uncertainty — the reliability of workers, AKG is still quite fast, and thus is suitable for online implementation.

In order to investigate how sensitive the prior for workers' reliability ρ_w is, we generate the workers' true reliability parameters from three different distributions, Beta$(10, 1)$, Beta$(2, 1)$, and Beta$(5, 2)$, and compare the performances of AKG between using the true generating distribution as the prior and using the generic Beta$(4, 1)$ as the prior. The results are plotted in Figure 2.6. As one can see from Figure 2.6, using the true generating distribution and generic Beta$(4, 1)$ prior lead to very similar performance in all three cases. Although there are some small differences between the two groups of curves, they are not significant as to

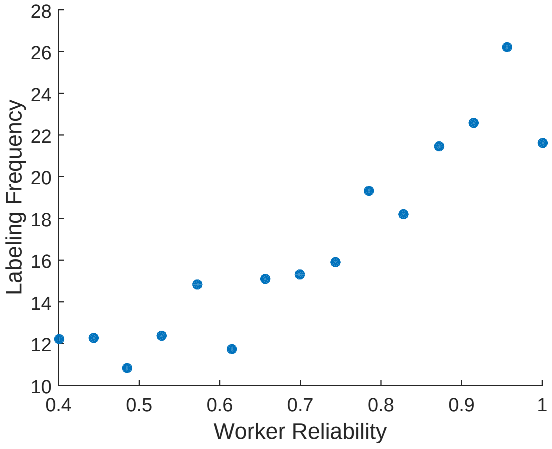
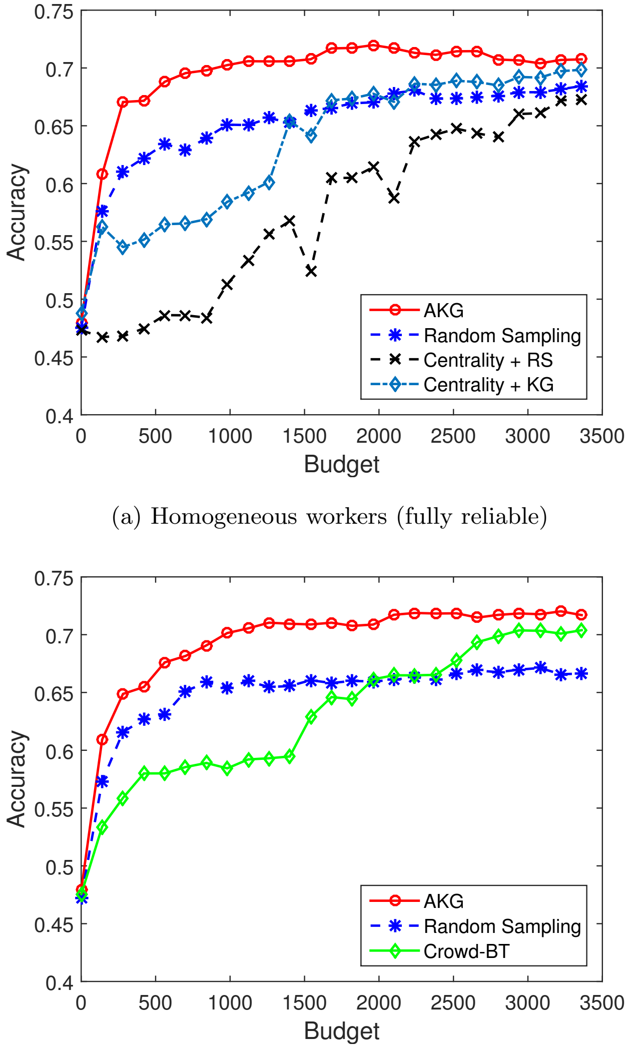


Figure 2.7: Averaged number of comparisons (a.k.a., labeling frequency) made by workers with different levels of reliability ρ_w.

the overall performance of the algorithm. This result shows that when there is no exact information on the quality of all workers, Beta$(4, 1)$ is a reasonable prior for workers' reliability and the proposed AKG policy is quite robust to the prior distribution in use.

Finally, we investigate whether good workers are indeed assigned more comparison tasks by our AKG policy in the setting of heterogeneous workers. In particular, we consider $K = 10$ items and $M = 15$ workers with the workers' true reliability parameters $\rho_w, w = 1, 2, \ldots, M$ ranging from 0.4 to 1 with an equal space in between. This crowd of workers is fixed and the total budget in each trial $T = 250$. We report the averaged number of pairs assigned to workers with different levels of reliability in Figure 2.7. As one can see from Figure 2.7, there is a clear trend that more reliable workers receive more pairs on average.

2.5.3 Real Data Study



(a) Homogeneous workers (fully reliable)

(b) Heterogeneous workers

Figure 2.8: Performance comparison on the real dataset

We now apply the proposed AKG policy (Algorithm 3) to a real dataset on reading difficulty levels [26]. The dataset comprises $K = 491$ different paragraphs, each assigned an integer-valued true reading difficulty score ranging from $1, 2, \ldots, 12$. Here, a higher score means the paragraph is more difficult to read. A total number of $M = 217$ different workers from Canada and the United States performed the comparison tasks on an online crowdsourcing platform called CrowdFlower[3]. Each worker was presented a pair of paragraphs every time and the worker identified which paragraph is more difficult to read. To overcome the issue of an imbalanced judgemental pool, each worker was allowed to compare at most 40 different pairs. There are

[3] http://www.crowdflower.com/

7,898 pairwise comparison results available in this dataset. Using these pairwise labels, we apply the AKG policy to recover the ranking by difficulty of these 491 paragraphs. We note that since the underlying truth is given as a difficulty level (1–12) for each paragraph (denoted by s_i for $i = 1, \ldots, K$) instead of a global ranking, we measure the accuracy of a ranking π as

$$\frac{2}{K(K-1)} \mathop{}_{i \neq j} \mathbf{1}_{\{\pi(i)>\pi(j)\}} \mathbf{1}_{\{s_i \geq s_j\}}.$$

In the above definition of ranking accuracy, when two paragraphs have the same reading difficulty level, any ranking between this pair will be treated as correct. It is also worth noting that, in the knowledge gradient step in (2.59), it is possible that the selected triplet (i_t, j_t, w_t) does not exist in the dataset (i.e., the worker w_t did not compare i_t and j_t in this data). Hence, in our implementation of AKG, we select the triplet in the dataset that maximizes the right-hand side of (2.59). We set the prior of $\boldsymbol{\theta}$ to be the uniform distribution on the simplex. This dataset also comes with a rating for each worker which measures the long-run performance of this worker on CrowdFlower. A higher rating implies a higher reliability of the worker. This dataset shows the averaged workers' rating is above 0.75. Thus, we still use Beta(4,1) as the prior on workers' reliability.

We run experiments in two different settings. The first one assumes that all workers are homogeneous and fully reliable. In this setting, we only need to select the next pair of paragraphs to compare but can randomly choose a worker to perform the comparison task. In this case, four algorithms are implemented (AKG policy (Algorithm 1), random sampling, rank centrality with the random sampling policy, and rank centrality with the knowledge gradient policy) and we report the averaged accuracy over 100 independent trials in Figure 2.8a to minimize the sampling effect of randomly selecting the next worker. The second experiment incorporates the heterogeneous reliability of workers so that the algorithms have to select both the pair to compare and the worker to perform the comparison task. In this case, three algorithms, AKG policy (Algorithm 3), random sampling and Crowd-BT, are implemented and the result is shown in Figure 2.8b. As one can see from these two plots, AKG outperforms the other methods in both settings, especially when the amount of budget is relatively low. As the budget level increases, the performance of Crowd-BT and rank centrality will eventually improve and achieve a similar accuracy as AKG.

2.6 Conclusion

In this chapter, we address the dynamic budget allocation problem in crowdsourced ranking. Using the Kendall's tau with respect to the true ranking as the measure of ranking accuracy, we formulate the problem of maximizing expected Kendall's tau by sequential comparisons into a Bayesian Markov decision process. To further address the computational challenges (especially, solving the NP-hard MAX-LOP) involved in the decision process, we propose an approximated knowledge gradient policy, which is not only computationally efficient but also achieves good performance as shown in the experimental sections.

We note that although this chapter focuses on the Bradley-Terry-Luce model [13, 76], it will be interesting to study the dynamic sampling in crowdsourced ranking for other ranking models such as permutation-based models (e.g., Mallows [80] and CPS [92] models) or stochastically transitive models [38, 99]). Meanwhile, theoretical bounds on posterior approximation errors are difficult to obtain and error propagation does exist during each iteration of the algorithm. In our future analysis we would like to quantify this error. Another interesting future direction is to incorporate the feature information of each item into the probabilistic model of the pairwise comparison results and develop a dynamic sampling policy that can further improve the ranking accuracy via modeling the feature information.

2.7 Appendix

In this section, we provide detailed proofs of some propositions in the chapter.

Proof. (**of Proposition 1**)

We will only show that (2.18) and (2.19) can be represented as (2.20) when $Y_{ij} = 1$. The proof for $Y_{ij} = -1$ is similar.

It is known that $\mathbb{E}\left[\theta_k|\boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha}')\right] = \frac{\alpha_k}{\alpha_0}$ and $\mathbb{E}\ \ \theta_k^2|\boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha}')\ \ = \frac{\alpha_k(\alpha_k+1)}{\alpha_0(\alpha_0+1)}$ for $k = 1, 2, \ldots, K$, which characterize the left-hand sides of (2.18) and (2.19).

With elementary calculus, we can show

$$\Pr(Y_{ij} = 1|\boldsymbol{\alpha}) = \underset{\Delta}{}\frac{\theta_i}{\theta_i + \theta_j}\frac{1}{\mathrm{B}(\boldsymbol{\alpha})}\overset{K}{\underset{k=1}{}}\theta_k^{\alpha_k - 1}\mathrm{d}\boldsymbol{\theta} = \frac{\alpha_i}{\alpha_i + \alpha_j} \tag{2.62}$$

so that

$$p(\boldsymbol{\theta}|Y_{ij} = 1, \boldsymbol{\alpha}) = \frac{p(\boldsymbol{\theta}, Y_{ij} = 1|\boldsymbol{\alpha})}{\Pr(Y_{ij} = 1|\boldsymbol{\alpha})} = \frac{\alpha_i + \alpha_j}{\alpha_i}\frac{\theta_i}{\theta_i + \theta_j}\frac{1}{\mathrm{B}(\alpha)}\overset{K}{\underset{k=1}{}}\theta_k^{\alpha_k - 1}. \tag{2.63}$$

Let $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_K)$ with $\beta_i = \alpha_i + 1$ and $\beta_k = \alpha_k$ for $k = i$. Then, we can show that

$$\begin{aligned}
\mathbb{E}[\theta_i|Y_{ij} = 1, \boldsymbol{\alpha}] &= \frac{\alpha_i + \alpha_j}{\alpha_i}\ \ \underset{\Delta}{}\frac{\theta_i^2}{\theta_i + \theta_j}\frac{1}{\mathrm{B}(\boldsymbol{\alpha})}\overset{K}{\underset{k=1}{}}\theta_k^{\alpha_k - 1}\mathrm{d}\boldsymbol{\theta} \\
&= \frac{\alpha_i + \alpha_j}{\alpha_0}\ \ \underset{\Delta}{}\frac{\theta_i}{\theta_i + \theta_j}\frac{1}{\mathrm{B}(\boldsymbol{\beta})}\overset{K}{\underset{k=1}{}}\theta_k^{\beta_k - 1}\mathrm{d}\boldsymbol{\theta} \\
&= \frac{\alpha_i + 1}{\alpha_0}\frac{\alpha_i + \alpha_j}{\alpha_i + \alpha_j + 1},
\end{aligned} \tag{2.64}$$

where the first and the third equalities are due to (2.62) and (2.63) and the second equality is by the definition of $\boldsymbol{\beta}$ and the property $\Gamma(x + 1) = x\Gamma(x)$ of Gamma function. Using a similar argument, we can show that

$$\mathbb{E}[\theta_j|Y_{ij} = 1, \boldsymbol{\alpha}] = \frac{\alpha_j}{\alpha_0}\frac{\alpha_i + \alpha_j}{\alpha_i + \alpha_j + 1} \tag{2.65}$$

$$\mathbb{E}[\theta_k|Y_{ij} = 1, \boldsymbol{\alpha}] = \frac{\alpha_k}{\alpha_0} \quad \text{for } k = i, j \tag{2.66}$$

$$\mathbb{E}\ \ \theta_i^2|Y_{ij} = 1, \boldsymbol{\alpha}\ \ = \frac{\alpha_i + 1}{\alpha_0}\frac{\alpha_i + 2}{\alpha_0 + 1}\frac{\alpha_i + \alpha_j}{\alpha_i + \alpha_j + 2} \tag{2.67}$$

$$\mathbb{E}\ \ \theta_j^2|Y_{ij} = 1, \boldsymbol{\alpha}\ \ = \frac{\alpha_j}{\alpha_0}\frac{\alpha_j + 1}{\alpha_0 + 1}\frac{\alpha_i + \alpha_j}{\alpha_i + \alpha_j + 2} \tag{2.68}$$

$$\mathbb{E}\ \ \theta_k^2|Y_{ij} = 1, \boldsymbol{\alpha}\ \ = \frac{\alpha_k}{\alpha_0}\frac{\alpha_k + 1}{\alpha_0 + 1} \quad \text{for } k = i, j. \tag{2.69}$$

Note that, when $Y_{ij} = 1$, the right-hand sides of (2.18) and (2.19) can be represented as the right-hand side of (2.20) using (2.64)∼(2.69) □

Proof. (**of Proposition 3**)

We will only show the conclusion when $Y_{ij}^w = 1$. The proof for $Y_{ij}^w = -1$ is similar.

When $Y_{ij}^w = 1$, we have $\eta_{ijw} = \frac{\mu_w\alpha_i}{\mu_w\alpha_i+\nu_w\alpha_j}$. We will first show (2.44) and (2.45) can be represented as the first four equations in (2.48). Since $\mathbb{E}\left[\theta_k|\boldsymbol{\theta}\sim \mathrm{Dir}(\boldsymbol{\alpha}')\right] = \frac{\alpha_k}{\alpha_0}$ and $\mathbb{E}\ \ \theta_k^2|\boldsymbol{\theta}\sim \mathrm{Dir}(\boldsymbol{\alpha}')\ \ = \frac{\alpha_k(\alpha_k+1)}{\alpha_0(\alpha_0+1)}$ for $k = 1, 2, \ldots, K$, the left-hand sides of the first four equations in (2.48) and those of (2.44) and (2.45) are identical.

With (2.62) and some basic properties of the Beta distribution, we can show

$$\begin{aligned}\Pr(Y_{ij}^w = 1|\boldsymbol{\alpha}, \mu_w, \nu_w) &= \mathbb{E}\ \ \rho_w\frac{\theta_i}{\theta_i+\theta_j} + (1-\rho_w)\frac{\theta_j}{\theta_i+\theta_j}\ \ \boldsymbol{\theta}\sim\mathrm{Dir}(\boldsymbol{\alpha}), \rho_w\sim\mathcal{B}(\mu_w,\nu_w) \\ &= \frac{\mu_w}{\mu_w+\nu_w}\frac{\alpha_i}{\alpha_i+\alpha_j} + \frac{\nu_w}{\mu_w+\nu_w}\frac{\alpha_j}{\alpha_i+\alpha_j} \qquad (2.70)\end{aligned}$$

so that

$$\begin{aligned}&p(\boldsymbol{\theta}, \rho_w|Y_{ij}^w = 1, \boldsymbol{\alpha}, \mu_w, \nu_w) \qquad (2.71)\\ =\ &\frac{\left(\rho_w\frac{\theta_i}{\theta_i+\theta_j} + (1-\rho_w)\frac{\theta_j}{\theta_i+\theta_j}\right)\frac{1}{\mathrm{B}(\boldsymbol{\alpha})\mathrm{B}(\mu_w,\nu_w)}\ \ {}_{k=1}^{K}\,\theta_k^{\alpha_k-1}\rho_w^{\mu_w-1}(1-\rho_w)^{\nu_w-1}}{\frac{\mu_w}{\mu_w+\nu_w}\frac{\alpha_i}{\alpha_i+\alpha_j} + \frac{\nu_w}{\mu_w+\nu_w}\frac{\alpha_j}{\alpha_i+\alpha_j}}.\end{aligned}$$

The equations (2.70) and (2.71), together with (2.64), imply

$$\begin{aligned}&\mathbb{E}[\theta_i|Y_{ij}^w = 1, \boldsymbol{\alpha}, \mu_w, \nu_w] \\ =\ &\frac{{}_0^1\ {}_\Delta\left(\rho_w\frac{\theta_i^2}{\theta_i+\theta_j} + (1-\rho_w)\frac{\theta_j\theta_i}{\theta_i+\theta_j}\right)\frac{1}{\mathrm{B}(\boldsymbol{\alpha})\mathrm{B}(\mu_w,\nu_w)}\ \ {}_{k=1}^{K}\,\theta_k^{\alpha_k-1}\rho_w^{\mu_w-1}(1-\rho_w)^{\nu_w-1}\mathrm{d}\boldsymbol{\theta}\mathrm{d}\rho_w}{\frac{\mu_w}{\mu_w+\nu_w}\frac{\alpha_i}{\alpha_i+\alpha_j} + \frac{\nu_w}{\mu_w+\nu_w}\frac{\alpha_j}{\alpha_i+\alpha_j}} \\ =\ &\frac{{}_\Delta\left(\frac{\mu_w}{\mu_w+\nu_w}\frac{\theta_i^2}{\theta_i+\theta_j} + \frac{\nu_w}{\mu_w+\nu_w}\frac{\theta_j\theta_i}{\theta_i+\theta_j}\right)\frac{1}{\mathrm{B}(\boldsymbol{\alpha})}\ \ {}_{k=1}^{K}\,\theta_k^{\alpha_k-1}\mathrm{d}\boldsymbol{\theta}}{\frac{\mu_w}{\mu_w+\nu_w}\frac{\alpha_i}{\alpha_i+\alpha_j} + \frac{\nu_w}{\mu_w+\nu_w}\frac{\alpha_j}{\alpha_i+\alpha_j}} \\ =\ &\frac{\frac{\mu_w}{\mu_w+\nu_w}\frac{\alpha_i}{\alpha_0}\frac{\alpha_i+1}{\alpha_i+\alpha_j+1}}{\frac{\mu_w}{\mu_w+\nu_w}\frac{\alpha_i}{\alpha_i+\alpha_j} + \frac{\nu_w}{\mu_w+\nu_w}\frac{\alpha_j}{\alpha_i+\alpha_j}} + \frac{\frac{\nu_w}{\mu_w+\nu_w}\frac{\alpha_i}{\alpha_0}\frac{\alpha_j}{\alpha_i+\alpha_j+1}}{\frac{\mu_w}{\mu_w+\nu_w}\frac{\alpha_i}{\alpha_i+\alpha_j} + \frac{\nu_w}{\mu_w+\nu_w}\frac{\alpha_j}{\alpha_i+\alpha_j}} \\ =\ &\eta_{ijw}\frac{(\alpha_i+1)(\alpha_i+\alpha_j)}{\alpha_0(\alpha_i+\alpha_j+1)} + (1-\eta_{ijw})\frac{\alpha_i(\alpha_i+\alpha_j)}{\alpha_0(\alpha_i+\alpha_j+1)}. \qquad (2.72)\end{aligned}$$

Using a similar argument, we can show that

$$\mathbb{E}[\theta_j|Y_{ij}^w = 1, \boldsymbol{\alpha}, \mu_w, \nu_w] = \eta_{ijw}\frac{\alpha_j(\alpha_i+\alpha_j)}{\alpha_0(\alpha_i+\alpha_j+1)} + (1-\eta_{ijw})\frac{(\alpha_j+1)(\alpha_i+\alpha_j)}{\alpha_0(\alpha_i+\alpha_j+1)} \qquad (2.73)$$

$$\mathbb{E}[\theta_k|Y_{ij}^w = 1, \boldsymbol{\alpha}, \mu_w, \nu_w] = \frac{\alpha_k}{\alpha_0} \quad \text{for } k = i, j \qquad (2.74)$$

$$\mathbb{E}\ \ \theta_i^2|Y_{ij}^w = 1, \boldsymbol{\alpha}, \mu_w, \nu_w\ \ = \frac{\eta_{ijw}(\alpha_i+1)(\alpha_i+2)(\alpha_i+\alpha_j)}{\alpha_0(\alpha_0+1)(\alpha_i+\alpha_j+2)} + \frac{(1-\eta_{ijw})\alpha_i(\alpha_i+1)(\alpha_i+\alpha_j)}{\alpha_0(\alpha_0+1)(\alpha_i+\alpha_j+2)} \qquad (2.75)$$

$$\mathbb{E}\ \ \theta_j^2|Y_{ij}^w = 1, \boldsymbol{\alpha}, \mu_w, \nu_w\ \ = \frac{\eta_{ijw}\alpha_j(\alpha_j+1)(\alpha_i+\alpha_j)}{\alpha_0(\alpha_0+1)(\alpha_i+\alpha_j+2)} + \frac{(1-\eta_{ijw})(\alpha_j+1)(\alpha_j+2)(\alpha_i+\alpha_j)}{\alpha_0(\alpha_0+1)(\alpha_i+\alpha_j+2)} \qquad (2.76)$$

$$\mathbb{E}\ \ \theta_k^2|Y_{ij}^w = 1, \boldsymbol{\alpha}, \mu_w, \nu_w\ \ = \frac{\alpha_k}{\alpha_0}\frac{\alpha_k+1}{\alpha_0+1} \quad \text{for } k = i, j. \qquad (2.77)$$

In the next, we will show (2.46) and (2.47) can be represented as the last two equations in (2.48). When $Y_{ij}^w = 1$, the last two equations in (2.48) become

$$\begin{cases}\frac{\mu_w}{\mu_w+\nu_w} &= \eta_{ijw}\frac{\mu_w+1}{\mu_w+\nu_w+1} + (1-\eta_{ijw})\frac{\mu_w}{\mu_w+\nu_w+1} \\ \frac{\mu_w(\mu_w+1)+\nu_w(\nu_w+1)}{(\mu_w+\nu_w)(\mu_w+\nu_w+1)} &= \eta_{ijk}\frac{(\mu_w+1)(\mu_w+2)}{(\mu_w+\nu_w+1)(\mu_w+\nu_w+2)} + (1-\eta_{ijk})\frac{(\mu_w)(\mu_w+1)}{(\mu_w+\nu_w+1)(\mu_w+\nu_w+2)} \\ &\quad +\eta_{ijk}\frac{(\nu_w)(\nu_w+1)}{(\mu_w+\nu_w+1)(\mu_w+\nu_w+2)} + (1-\eta_{ijk})\frac{(\nu_w+1)(\nu_w+2)}{(\mu_w+\nu_w+1)(\mu_w+\nu_w+2)}.\end{cases} \qquad (2.78)$$

It is known that $\mathbb{E}\left[\rho_w|\rho_w \sim \mathcal{B}(\mu'_w,\nu'_w)\right] = \frac{\mu_w}{\mu_w+\nu_w}$, $\mathbb{E}\ \ \rho_w^2|\rho_w \sim \mathcal{B}(\mu'_w,\nu'_w)\ \ = \frac{\mu_w(\mu_w+1)}{(\mu_w+\nu_w)(\mu_w+\nu_w+1)}$ and $\mathbb{E}\ \ (1-\rho_w)^2|\rho_w \sim \mathcal{B}(\mu'_w,\nu'_w$ $\frac{\nu_w(\nu_w+1)}{(\mu_w+\nu_w)(\mu_w+\nu_w+1)}$, indicating that the left-hand sides of (2.46) and (2.47) match those of (2.78).

To characterize the right-hand sides of (2.46) and (2.47), we first derive from (2.71) that

$$
\begin{aligned}
&\mathbb{E}\ \ \rho_w|Y^w_{ij},\boldsymbol{\alpha},\boldsymbol{\mu},\boldsymbol{\nu}\\
=\ &\frac{{}^{1}_{0}\ \ {}_{\Delta}\ \ \rho_w^2\frac{\theta_i}{\theta_i+\theta_j}+\rho_w(1-\rho_w)\frac{\theta_j}{\theta_i+\theta_j}\quad \frac{1}{\mathrm{B}(\boldsymbol{\alpha})\mathrm{B}(\mu_w,\nu_w)}\quad {}^{K}_{k=1}\,\theta_k^{\alpha_k-1}\rho_w^{\mu_w-1}(1-\rho_w)^{\nu_w-1}\mathrm{d}\boldsymbol{\theta}\mathrm{d}\rho_w}{\frac{\mu_w}{\mu_w+\nu_w}\frac{\alpha_i}{\alpha_i+\alpha_j}+\frac{\nu_w}{\mu_w+\nu_w}\frac{\alpha_j}{\alpha_i+\alpha_j}}\\
=\ &\frac{{}^{1}_{0}\ \ \frac{\alpha_i}{\alpha_i+\alpha_j}\rho_w^2+\frac{\alpha_j}{\alpha_i+\alpha_j}\rho_w(1-\rho_w)\quad \frac{1}{\mathrm{B}(\mu_w,\nu_w)}\rho_w^{\mu_w-1}(1-\rho_w)^{\nu_w-1}\mathrm{d}\rho_w}{\frac{\mu_w}{\mu_w+\nu_w}\frac{\alpha_i}{\alpha_i+\alpha_j}+\frac{\nu_w}{\mu_w+\nu_w}\frac{\alpha_j}{\alpha_i+\alpha_j}}\\
=\ &\frac{\frac{\mu_w}{\mu_w+\nu_w}\frac{\mu_w+1}{\mu_w+\nu_w+1}\frac{\alpha_i}{\alpha_i+\alpha_j}}{\frac{\mu_w}{\mu_w+\nu_w}\frac{\alpha_i}{\alpha_i+\alpha_j}+\frac{\nu_w}{\mu_w+\nu_w}\frac{\alpha_j}{\alpha_i+\alpha_j}}+\frac{\frac{\mu_w}{\mu_w+\nu_w}\frac{\nu_w}{\mu_w+\nu_w+1}\frac{\alpha_j}{\alpha_i+\alpha_j}}{\frac{\mu_w}{\mu_w+\nu_w}\frac{\alpha_i}{\alpha_i+\alpha_j}+\frac{\nu_w}{\mu_w+\nu_w}\frac{\alpha_j}{\alpha_i+\alpha_j}}\\
=\ &\eta_{ijw}\frac{\mu_w+1}{\mu_w+\nu_w+1}+(1-\eta_{ijw})\frac{\mu_w}{\mu_w+\nu_w+1}. \qquad (2.79)
\end{aligned}
$$

Following a similar procedure, we can show

$$
\begin{aligned}
&\mathbb{E}[\rho_w^2+(1-\rho_w)^2|o_i\succ_w o_j,\theta\sim Dir(\alpha),\rho_w\sim Beta(\mu_w,\nu_w)]\\
=\ &\frac{\frac{\mu_w}{\mu_w+\nu_w}\frac{\mu_w+1}{\mu_w+\nu_w+1}\frac{\mu_w+2}{\mu_w+\nu_w+2}\frac{\alpha_i}{\alpha_i+\alpha_j}}{\frac{\mu_w}{\mu_w+\nu_w}\frac{\alpha_i}{\alpha_i+\alpha_j}+\frac{\nu_w}{\mu_w+\nu_w}\frac{\alpha_j}{\alpha_i+\alpha_j}}+\frac{\frac{\mu_w}{\mu_w+\nu_w}\frac{\mu_w+1}{\mu_w+\nu_w+1}\frac{\nu_w}{\mu_w+\nu_w+2}\frac{\alpha_j}{\alpha_i+\alpha_j}}{\frac{\mu_w}{\mu_w+\nu_w}\frac{\alpha_i}{\alpha_i+\alpha_j}+\frac{\nu_w}{\mu_w+\nu_w}\frac{\alpha_j}{\alpha_i+\alpha_j}}\\
&+\frac{\frac{\nu_w}{\mu_w+\nu_w}\frac{\nu_w+1}{\mu_w+\nu_w+1}\frac{\mu_w}{\mu_w+\nu_w+2}\frac{\alpha_i}{\alpha_i+\alpha_j}}{\frac{\mu_w}{\mu_w+\nu_w}\frac{\alpha_i}{\alpha_i+\alpha_j}+\frac{\nu_w}{\mu_w+\nu_w}\frac{\alpha_j}{\alpha_i+\alpha_j}}+\frac{\frac{\nu_w}{\mu_w+\nu_w}\frac{\nu_w+1}{\mu_w+\nu_w+1}\frac{\nu_w+2}{\mu_w+\nu_w+2}\frac{\alpha_j}{\alpha_i+\alpha_j}}{\frac{\mu_w}{\mu_w+\nu_w}\frac{\alpha_i}{\alpha_i+\alpha_j}+\frac{\nu_w}{\mu_w+\nu_w}\frac{\alpha_j}{\alpha_i+\alpha_j}}\\
=\ &\eta_{ijw}\frac{(\mu_w+1)(\mu_w+2)}{(\mu_w+\nu_w+1)(\mu_w+\nu_w+2)}+(1-\eta_{ijw})\frac{(\mu_w)(\mu_w+1)}{(\mu_w+\nu_w+1)(\mu_w+\nu_w+2)} \qquad (2.80)\\
&+\eta_{ijw}\frac{(\nu_w)(\nu_w+1)}{(\mu_w+\nu_w+1)(\mu_w+\nu_w+2)}+(1-\eta_{ijw})\frac{(\nu_w+1)(\nu_w+2)}{(\mu_w+\nu_w+1)(\mu_w+\nu_w+2)}.
\end{aligned}
$$

Putting (2.79) and (2.80) together, we have shown that the right-hand sides of (2.78) are exactly the right-hand sides of (2.70) and (2.71), which completes the proof. □

Chapter 3

Retail

3.1 Introduction

Retailers routinely collect large volumes of historical data. These data are used to improve future business practices such as inventory management, pricing decisions, and customer segmentation. One of the most important data-driven task for a retailer is to predict the demand for each stock keeping unit (SKU). A common approach in practice is to split the SKUs in departments (e.g., soft drinks), and sometimes even in sub-categories (e.g, a specific format of soft drinks).

Predictive models for demand prediction have been extensively studied and applied in practice. A typical demand model is a regression specification with the sales (or logarithmic of the sales) as the outcome variable, and price, seasonality, brand, color, and promotion as independent variables or features. The model coefficients are then estimated using historical data.

In many retail settings, a subset of items have been offered for a long time, whereas other items were recently introduced. For such newly introduced items, only a limited number of historical observations is available. It is thus crucial to reduce the dimensionality of the feature space in order to decrease the variance of the estimated coefficients. At the same time, the SKUs in the same department often share similar characteristics, and hence tend to be affected by a particular feature in a similar way. A prominent approach is to estimate certain coefficients at the aggregate level (i.e., by gathering the data across all SKUs and assuming a uniform coefficient). For example, it seems reasonable to believe that all the items in the ice-cream category share the same seasonality pattern. Although this approach has been widely adopted in the retail industry, no rigorous empirical method has been developed to formalize how this data aggregation procedure should be applied for demand prediction. In this chapter, we seek to bridge this gap by formalizing the tradeoff between data aggregation (i.e., finding the right level of aggregation for each coefficient) and model flexibility (i.e., estimating a different model for each item) in a systematic fashion.

Due to insufficient data, the traditional approach of estimating a different model for each SKU is usually inefficient for new products or for SKUs with noisy observations. This approach cannot identify the right aggregation level for each coefficient, and does not find the underlying cluster structure of the coefficients. Based on common clustering methods (e.g., k-means), we propose an efficient and integrated approach to infer the coefficient of each feature while dynamically identifying the right level of data aggregation based on the statistical properties of the estimated coefficients. Our method also allows us to incorporate multiple aggregation levels while preserving model interpretability. From a practical perspective, our method can be easily estimated using retail data and yields a significant improvement in out-of-sample prediction accuracy.

3.1.1 Main Results and Contributions

We study the tradeoff between data aggregation and model flexibility by optimally identifying the right level of aggregation for each feature as well as the cluster structure of the items. We propose a practical method—referred to as the Dynamic Aggregation with Clustering (DAC) algorithm—that allows us to predict demand while optimally identifying the features that should be estimated at the (i) item, (ii) cluster, and (iii) aggregated levels. Our proposed algorithm first applies the maximum likelihood estimation approach to estimate a different coefficient vector for each item (called the decentralized model). It then performs a hypothesis test (i.e., $t-$test) on the estimated coefficients from the decentralized model to identify the right aggregation level of each feature. To characterize the cluster structure of the items, we apply the $k-$means method on the estimated coefficients from the decentralized model (as opposed to the features themselves).

We first characterize the theoretical properties of the DAC. Specifically, we show that the DAC yields a consistent estimate of the (i) data aggregation levels, (ii) cluster structure, and (iii) feature coefficients. As a result, if the data has enough observations, one can correctly identify the "true" data generating process. In addition to this consistency result, we demonstrate improved asymptotic properties—smaller variance and a tighter probabilistic bound—relative to the commonly used ordinary least squares method. Armed with these theoretical results, we next conduct several computational experiments—based on both simulated and real data—to illustrate the significant improvement of the DAC in (out-of-sample) prediction accuracy relative to several common benchmarks. Our results highlight the essential value of the proposed DAC algorithm in better balancing the bias-variance tradeoff, resulting in more accurate demand prediction. Finally, we apply the DAC using two years of retail data and convey that it can also help retailers discover useful insights on the relationships between the different items.

3.1.2 Related Literature

This chapter is related to several streams of literature including (i) prediction and clustering algorithms and (ii) retail operations and demand forecasting.

Prediction and clustering algorithms: The problems of demand prediction and clustering are extensively studied in the machine learning (ML) literature. [10] combine ideas from ML and operations research to develop a new prediction method. The authors solve a conditional stochastic optimization problem by incorporating ML methods such as local regression and random forests. [62] and [33] focus on developing new ML methods by training a prediction model with respect to a nominal optimization problem. Although several previous papers study general settings, it is hard to apply existing methods to a retail setting where multiple levels of hierarchy may exist. [34] propose a new idea called Smart "Predict, then Optimize" (SPO). The key component of SPO is that the loss function is computed based on comparing the objective values generated using predicted and observed data. The authors then address the computational challenge and develop a tractable version of SPO.

Our work is also related to the traditional clustering literature. Since the introduction of k-means by [78], clustering algorithms have been extensively studied. In particular, k-means has been widely applied to a variety of domains, such as image segmentation [81]. In this chapter, we leverage some theoretical properties of k-means and embed this clustering method as one of the key steps in our algorithm.

Retail operations and demand forecasting: Retailers always seek to improve operational decisions, such as inventory replenishment, supply chain management, and revenue management and pricing. These decisions all closely rely on accurate demand forecasting/prediction. There is a large body of literature that focuses on developing methods for demand prediction in retail settings. Given the increasing volume of transaction data collected by retailers, sophisticated models have emerged in the past two decades. Marketing papers such as [39] and [28] estimate econometrics models to draw managerial insights on the impact of retail promotions. In a similar vein, [107] and [77] study the pre- and post-promotion dips using linear regression models with lagged variables. Recent developments in demand prediction include the following three papers: [56] try to embed the competitive information (including price and promotions) into demand prediction, [37] suggest that promotional information can be quite valuable in improving forecast accuracy, and [57] further take into account the impact of marketing activities. In the operations management community, demand prediction models are often used as an input to an optimization problem. For example, [25] estimate a log-log demand model using supermarket data. The authors then solve the promotion optimization problem by developing an approximation based on linear programming.

A recent stream of papers integrate a clustering step into demand prediction. For instance, [6] propose an iterative approach to cluster products and leverage existing data to predict the sales of new products. [55] propose a two-step approach to first estimate the product lifecycle, and then cluster and predict. In this chapter, however, the definition of clusters is fundamentally different. Unlike previous work, our clustering is based on the estimated coefficients rather than on the features. Furthermore, our model is flexible enough to account for different levels of data aggregation, whereas in previous studies, all features are essentially estimated at the cluster level. Allowing such flexibility is key to improve demand forecasting.

Structure of the chapter. The rest of the chapter is organized as follows. In Section 3.2, we introduce our model and discuss the relevant computational challenges. We then describe the DAC algorithm in Section 3.3. Our analytical results are presented in Section 3.4. In Sections 3.5 and 3.6, we conduct computational experiments using simulated and real data, respectively. Our conclusions are reported in

Section 4.7. The proofs of our analytical results are relegated to the Appendix.

3.2 Model

We introduce our demand prediction model under the generalized linear model (GLM) framework. We consider a retail department (e.g., soft drinks, electronics) which comprises n items (or SKUs). Each item has m historical observations (e.g., weekly sales information). We use $Y_{i,j}$ to denote the (log-)sales of item i in observation j ($1 \leq i \leq n$ and $1 \leq j \leq m$). The prediction model of each item includes d (for simplicity of exposition, we assume that each item has the same number of features and observations). The feature set is denoted by $D := \{1, 2, \cdots, d\}$. We also define $X_{i,j} := (X^1_{i,j}, X^2_{i,j}, \cdots, X^d_{i,j})' \in \mathbb{R}^d$ as the feature vector for item i and observation j.

An important characteristic of our model is that a feature $l \in D$ may affect the demand/sales of an item at different aggregation levels: (i) Department, (ii) SKU, and (iii) Cluster. More precisely, a feature may have the same impact on all items, captured by a uniform coefficient for all the items in the department. We refer to such features as *shared* (department-level) features, the set of which is denoted by D_s. Alternatively, a feature may have a different impact for different items, captured by different coefficients for different items. We refer to such features as *non-shared* (SKU-level) features, the set of which is denoted by D_n. Finally, we assume that the items follow a cluster structure so that some features have the same impact for items within the same cluster and a different impact for items in a different cluster. This phenomenon is captured by a uniform coefficient for all the items in the same cluster (the coefficients are different across different clusters). We refer to such features as *cluster* (cluster-level) features, the set of which is denoted by D_c. We also assume that the number of clusters k is given but the cluster structure is unknown (one can further apply our proposed algorithm for different values of k). The entire feature set, D, can be written as the union of three disjoint sets of features that affect the demand at different aggregation levels: $D = D_s \cup D_n \cup D_c$. The feature aggregation structure D_s, D_n, and D_c are unknown a priori and should be estimated from data. The underlying cluster structure is also unknown.

In the GLM framework, the observations are generated from an exponential family distribution which includes normal, binomial, and Poisson distributions as special cases. Based on the three aggregation levels of the features, we have:

$$\mathbb{E}[Y_{i,j}] = g^{-1}\left(\sum_{l \in D_s} X^l_{i,j}\beta^s_l + \sum_{l \in D_n} X^l_{i,j}\beta^n_{i,l} + \sum_{l \in D_c} X^l_{i,j}\beta^c_{\mathcal{C}(i),l}\right), \qquad i = 1, \ldots, n \text{ and } j = 1, \ldots, m. \tag{3.1}$$

Here, $\mathcal{C}(i) \in \{1, \ldots, k\}$ is the cluster that contains item i and $g(\cdot)$ represents the link function that establishes the relationship between the linear predictor and the mean of the outcome variable. Furthermore, we use $\mathcal{C}_u$ to denote items in cluster u, where $u \in \{1, 2, \ldots, k\}$ and $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k\}$ is a partition of the items $\{1, 2, \ldots, n\}$. There are many commonly used link functions, and in practice the choice is made depending on the setting. For example, if $Y_{i,j}$ is the number of units sold of item i on observation j, $g(\cdot)$ could be the identity function and, as a result, the model reduces to a linear regression. On the other hand, if $Y_{i,j}$ is a binary variable, $g(\cdot)$ can be a logit function. Likewise, there are other examples of link functions such as logarithmic and inverse squared.

Based on Equation (3.1), we can characterize the aggregation levels of the three types of features. For a department-level feature l, its coefficient β^s_l is shared among all items. In other words, all items in the department have the same coefficient for this feature. In comparison, for a SKU-level feature l, its coefficient $\beta^n_{i,l}$ varies across the different items (i.e., $\beta^n_{i,l} = \beta^n_{k,l}$ for $i = k$). Finally, for a cluster-level feature l, all items in the same cluster will have the same coefficient, that is, if $i \in \mathcal{C}(i)$, then the coefficient of $X^l_{i,j}$ is equal to $\beta^c_{\mathcal{C}(i),l}$. Thus, the total number of coefficients in our model is $d_x = n|D_n| + k|D_c| + |D_s|$. Note that the notion of estimating the coefficient of certain features at an aggregated level is common in practice. For example, retailers sometimes estimate the seasonality coefficients at the department level to avoid over-fitting. Also, when estimating the effects of promotions such as cannibalization or halo, one would consider clustering some items together because promotions often have a similar impact on a group of items. For expositional and computational convenience, we make the following assumption throughout the chapter.

(a) If a feature l is at the SKU level (i.e., $l \in D_n$), then $\beta_{i,l} = \beta_{i\ ,l}$ for any two items i and i', that is, a SKU-level feature has a different effect for different items.

(b) If cluster-level features exist (i.e., $l \in D_c = \emptyset$), then $\beta_{i,l} = \beta_{i\ ,l}$ if and only if $\mathcal{C}(i) = \mathcal{C}(i')$, that is, a cluster-level feature has the same effect for items in the same cluster and a different effect for different clusters. Furthermore, each cluster has at least two items.

As we discuss below, our method can easily be adapted to the setting where the above assumptions are relaxed. We use $n_{i,l}$ to denote the number of items that share the same coefficient as item i for feature l, that is, $n_{i,l} = 1$ if $i \in D_n$, $n_{i,l} = n$ if $i \in D_s$, and $n_{i,l} = |\mathcal{C}(i)|$ if $i \in D_c$.

Our main goal is to accurately predict the dependent variable Y given the features X, assuming the data generating process in Equation (3.1). The key challenge lies in correctly estimating three essential aspects of our model: (a) the aggregation level of each feature, (b) the cluster structure, and (c) the coefficient of each feature. Before presenting our proposed estimation method, we first discuss why directly estimating the aggregation levels, cluster structure, and feature coefficients can be challenging. Two intuitive methods come to mind for estimating our model. First, one can use the *constrained maximum likelihood estimation* (constrained MLE). This approach revises the standard MLE by adding the constraint that, for items in the same cluster, the coefficients of the features at the department or cluster levels should be the same. Since we do not know a priori the aggregation level of each feature, the constrained MLE approach will involve solving an optimization problem with non-convex (multiplicative) constraints, which is computationally prohibitive when the number of features is large. We provide a more detailed discussion of the impracticality of this approach for our problem in Appendix 3.8.1.

A second possible approach is via *iterative optimization.* This approach introduces a binary decision variable to determine the aggregation level of each feature. To simultaneously estimate the aggregation level and the coefficient of the features, we iteratively estimate the aggregation level binary variable and the coefficients using a MLE. The iterative procedure will stop once the binary variables remain unchanged for two consecutive iterations. A similar iterative optimization approach was proposed by [6] to address the demand forecasting problem with two feature aggregation levels (SKU and cluster levels). In their setting, this iterative procedure was proved to converge to the true coefficients and aggregation levels (i.e., the estimate is consistent). In our setting, however, the validity of the iterative optimization approach heavily relies on the initialization of the parameters. Depending on the initial parameter, the procedure may reach a local optimum without any guarantee of convergence to a global optimal solution. For more details on the iterative optimization approach in our context, see Appendix 3.8.1. Finally, an important shortcoming of the constrained MLE and the iterative optimization is that neither method can simultaneously identify the cluster structure and estimate the coefficients.

3.3 Dynamic Data Aggregation with Clustering

As mentioned, the problem of estimating the feature coefficients, aggregation levels, and cluster structure is computationally challenging and subject to substantial prediction errors. In this section, we propose a novel dynamic data aggregation approach which allows us to (a) identify the right level of aggregation for each feature, (b) find the underlying cluster structure, and (c) generate a consistent estimate of the coefficients for the GLM model. Our method is entirely data-driven and could efficiently achieve the aforementioned three goals in an integrated fashion as long as we have sufficiently many observations in the training set (i.e., m is sufficiently large).

We start our analysis by focusing on a (simple) special case of the model in Equation (3.1). Specifically, we assume that all the features are at the SKU-level. In this case, the data generating process can be written as follows:

$$\mathbb{E}[Y_{i,j}] = g^{-1} \quad \underset{l\in D}{} X^l_{i,j} b_{i,l} \quad , \qquad i = 1,\ldots,n \text{ and } j = 1,\ldots,m. \tag{3.2}$$

Comparing Equations (3.1) and (3.2), we have $b_{i,l} = \beta^s_l$ for $l \in D_s$, $b_{i,l} = \beta^n_{i,l}$ for $l \in D_n$, and $b_{i,l} = \beta^c_{\mathcal{C}(i),l}$ for $l \in D_c$. We refer to model (3.2) as the *decentralized model*. The estimation of the decentralized model is usually carried out through iterative re-weighted least squares, which ultimately lead to the MLE. As expected, the estimation of the decentralized model can be decomposed into estimating each item separately. Specifically, using the data of item i, we apply the MLE to obtain the estimated coefficients of item i, $\hat{b}_i := (\hat{b}_{i,1}, \hat{b}_{i,2}, \cdots, \hat{b}_{i,d})' \in \mathbb{R}^d$, as follows:

$$\hat{b}_i \in \arg\max_{b_i} \quad \overset{m}{\underset{j=1}{}} \log \mathcal{L}(b_i | Y_{i,j}, X^1_{i,j}, X^2_{i,j}, \ldots, X^d_{i,j}),$$

where $\mathcal{L}(b_i | Y_{i,j}, X^1_{i,j}, X^2_{i,j}, \ldots, X^d_{i,j})$ is the likelihood function associated with the data $\{Y_{i,j}, (X^1_{i,j}, X^2_{i,j}, \ldots, X^d_{i,j})\}$ and the coefficient vector $b_i = (b_{i,1}, b_{i,2}, \cdots, b_{i,d})' \in \mathbb{R}^d$. We refer to the estimator $\hat{b} := (\hat{b}_1, \hat{b}_2, \cdots, \hat{b}_n)$ as the *decentralized estimator*. To estimate the aggregation levels, cluster structure, and feature coefficients, we need to find a partition of the vector $\hat{b}_i$ for each item i, to identify the correct level of aggregation for each feature. Before presenting our algorithm in greater detail, we first state the following consistency property of the decentralized estimator $\hat{b}$.

Lemma 4. *The decentralized estimator $\hat{b}$ is consistent, that is, if $m \uparrow +\infty$, we have:*

- $\hat{b}_{i,l} \xrightarrow{p} \beta^s_l$ *for* $l \in D_s$*;*
- $\hat{b}_{i,l} \xrightarrow{p} \beta^n_{i,l}$ *for* $l \in D_n$*;*
- $\hat{b}_{i,l} \xrightarrow{p} \beta^c_{\mathcal{C}(i),l}$ *for* $l \in D_c$*;*

where $\xrightarrow{p}$ *refers to convergence in probability.*

Lemma 4 shows that with sufficiently many observations, we can consistently estimate the feature coefficients using the decentralized MLE. Two issues remain unaddressed with the decentralized estimation: how can we find the right aggregation level for each feature and how can we identify the cluster structure of the items. Furthermore, the decentralized estimator may suffer from overfitting so that it may have a high variance. This follows from the fact that the number of coefficients of the true model—Equation (3.1)—is strictly less than the number of coefficients generated by the decentralized estimator: $d_x = n|D_n| + k|D_c| + |D_s| < nd$, where $d = |D_n| + |D_c| + |D_s|$.

It is not surprising that the decentralized estimator, $\hat{b}$, is consistent given that the decentralized model has the highest flexibility. As a result, if we have sufficiently many observations for each item, the forecast performance of the decentralized model will be reasonably good. That said, the decentralized model neither captures the aggregation level of each feature nor leverages the cluster structure of the items. As we discuss in Section 3.5, exploiting the data aggregation can substantially increase the prediction accuracy. Namely, data aggregation helps us reduce the variance of the estimator and addresses the over-fitting issue.

To estimate the aggregation level and the underlying cluster structure based on Equation (3.1), we next introduce an additional special case of the model in which the data aggregation level and cluster structure are known. We refer to this case as the *aggregated model* and we call its MLE the *aggregated estimator*, which we denote as $\hat{\beta}$. For the aggregated model, we denote $\hat{\beta}^s_l$ as the estimated coefficient for a department-level feature, $\hat{\beta}^n_{i,l}$ for a SKU-level feature, and $\hat{\beta}^c_{\mathcal{C}(i),l}$ for a cluster-level feature. We are now ready to introduce the *Dynamic Aggregation with Clustering* (DAC) algorithm, which allows us to consistently estimate the coefficient of each feature for each item, as well as correctly identify the right aggregation levels and the underlying cluster structure (see Algorithm 3).

```
Algorithm 3 DAC
Input: Estimated coefficient $\hat{b}_{i,l}$ and standard error ($S\hat{E}_{i,l}$) for each item $i$ and feature $l$.
For each feature $l \in D$:
    Fix an item 1. For all other items $i = 1$, compute the $p-$value based on the null hypothesis $H^0_{1,i}$ that $b_{1,l} = b_{i,l}$, that is, the coefficients of feature $l$ are the same for item 1 and item $i$.
    If $H^0_{1,i}$ is not rejected for all items, then feature $l$ should be estimated at the aggregated level.
    If $H^0_{1,i}$ is rejected for some items and validated for others, then feature $l$ should be estimated at the cluster level. We then run a one-dimensional $k$-means algorithm on $\{\hat{b}_{i,l} : 1 \leq i \leq n\}$ and obtain the resulting clusters $\hat{\mathcal{C}}_1, \hat{\mathcal{C}}_2, \cdots, \hat{\mathcal{C}}_k$.
    If $H^0_{1,i}$ is rejected for all items, then feature $l$ should be estimated at the SKU level.
    Obtain the aggregation level for each feature: $\hat{D}_n$, $\hat{D}_s$, and $\hat{D}_c$.
    Fit an aggregated model to obtain the coefficients $\hat{\beta}$.
Output: (a) Aggregation levels: $(\hat{D}_n, \hat{D}_s, \hat{D}_c)$, (b) cluster structure: $(\hat{\mathcal{C}}_1, \hat{\mathcal{C}}_2, \cdots, \hat{\mathcal{C}}_k)$, (c) Feature coefficients: $\hat{\beta}$.
```

The DAC is inspired by Lemma 4. By leveraging the consistent estimated parameters obtained from the decentralized model, we can perform a hypothesis testing to identify the right data aggregation levels and cluster structure. Options for hypothesis testing include the Wald test and the likelihood ratio test (see e.g., [47]). The main idea is that if the estimated coefficients $\hat{b}_{i,l}$ and $\hat{b}_{i\ ,l}$ are statistically close one another, then it is very likely that either item i and item i' belong to the same cluster or that feature l is an aggregated-level feature. Another interesting characteristics of our method is that it uses the estimated coefficients as inputs to identify the cluster structure of the items (as opposed to item attributes as in traditional clustering algorithms). If some clusters do not agree for different features, one can implement a majority vote to decide the optimal cluster structure. Alternatively, one can pool cluster-level feature coefficients and fit a multi-dimensional k-means. Either approach could produce a consistent estimate of the cluster structure. We note that under the assumptions, the pairwise hypothesis testing step has a time complexity of $O(n)$.[1] Furthermore, the identification of cluster-level features is even more efficient. Indeed, if we infer that the coefficients of a feature coincide for some items and differ for others, then this feature must be at the cluster-level.

We next show that DAC can consistently identify the aggregation level of each feature and the underlying cluster structure, under sufficiently many observations.

Proposition 5. *DAC outputs a consistent estimate of the aggregation level for each feature and of the underlying cluster structure of the items, that is,*

$$\lim_{m\uparrow\infty} \mathbb{P}\Big[(\hat{D}_n, \hat{D}_s, \hat{D}_c) = (D_n, D_s, D_c) \text{ or } (\hat{\mathcal{C}}_1, \hat{\mathcal{C}}_2, \cdots, \hat{\mathcal{C}}_k) \text{ is not a permutation of } (\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_k)\Big] = 0.$$

[1] If we relax the assumptions, the DAC can easily be adapted to run $O(n^2)$ hypothesis tests instead of $O(n)$.

As shown in Appendix 3.8.2, the main idea behind the proof of Proposition 5 is to leverage the consistency of the decentralized estimator. The estimated coefficients in the decentralized model will eventually converge to their true values, and hence allow us to accurately learn the aggregation level and the cluster structure.

3.4 Theoretical Properties of DAC

Since the decentralized model also produces a consistent estimator, the following question arises: What is the benefit of performing the pairwise tests and the clustering algorithm relative to the decentralized model? We address this question from three perspectives: (a) Analytical comparison between the aggregated and decentralized models, which highlights the value of data aggregation and cluster structure; (b) Simulation studies of DAC versus several benchmarks, which show that DAC can successfully identify and leverage the data aggregation and cluster structure; and (c) Implementation of DAC using retail data, which showcases the practical value of DAC in improving demand prediction accuracy. In this section, we examine the value of data aggregation and cluster structure from a theoretical perspective by showing several benefits of the aggregated model relative to the decentralized model.

To convey the benefits of DAC, we first observe that if the true data generating process has different aggregation levels for different features, though flexible, the decentralized model assumes an overly complex model and, hence, will be prone to over-fitting. To formalize this intuition, we leverage the asymptotic normality property of the MLE. Specifically, we denote $\mathcal{I}(\beta)$ as the Fisher's information matrix, which is the Hessian of the log-likelihood evaluated at the true coefficients:

$$\mathcal{I}(\beta) := \text{Hess}\left[\sum_{i=1}^{n}\sum_{j=1}^{m} \log \mathcal{L}(\beta | Y_{i,j}, X_{i,j}^1, \ldots, X_{i,j}^d)\right],$$

where $\beta \in \mathbb{R}^{d_x}$ is the true coefficient vector associated with the true data aggregation and cluster structure. Similarly, we denote $\mathcal{I}_i(\beta_i)$ as the Fisher's information matrix, which is the Hessian of the log-likelihood evaluated at the true coefficients for item i:

$$\mathcal{I}_i(\beta_i) := \frac{\partial^2}{\partial \beta_{i,l} \partial \beta_{i,l}}\left[\sum_{j=1}^{m} \log \mathcal{L}(\beta_i | Y_{i,j}, X_{i,j}^1, \ldots, X_{i,j}^d)\right],$$

where $\beta_i \in \mathbb{R}^d$ is the true coefficient vector associated with item i. Note that to obtain $\mathcal{I}(\beta)$, we need to specify the data aggregation and the cluster structure, which are not necessary to compute $\mathcal{I}_i(\beta_i)$. We are now ready to compare the (asymptotic) variances of the aggregated and decentralized models. We use $\text{Var}(\cdot)$ to denote the variance operator.

Proposition 6. *For the aggregated and decentralized models, the following statements hold:*

(a) $\hat{\beta}$ and $\hat{b}$ converge to the following asymptotic distributions as $m \to \infty$,

$$\begin{aligned} \sqrt{m}(\hat{\beta} - \beta) &\xrightarrow{d} N(0, \mathcal{I}(\beta)^{-1}), \\ \sqrt{m}(\hat{b}_i - \beta_i) &\xrightarrow{d} N(0, \mathcal{I}_i(\beta_i)^{-1}), \ \textit{for } i = 1, 2, \ldots, n, \end{aligned}$$

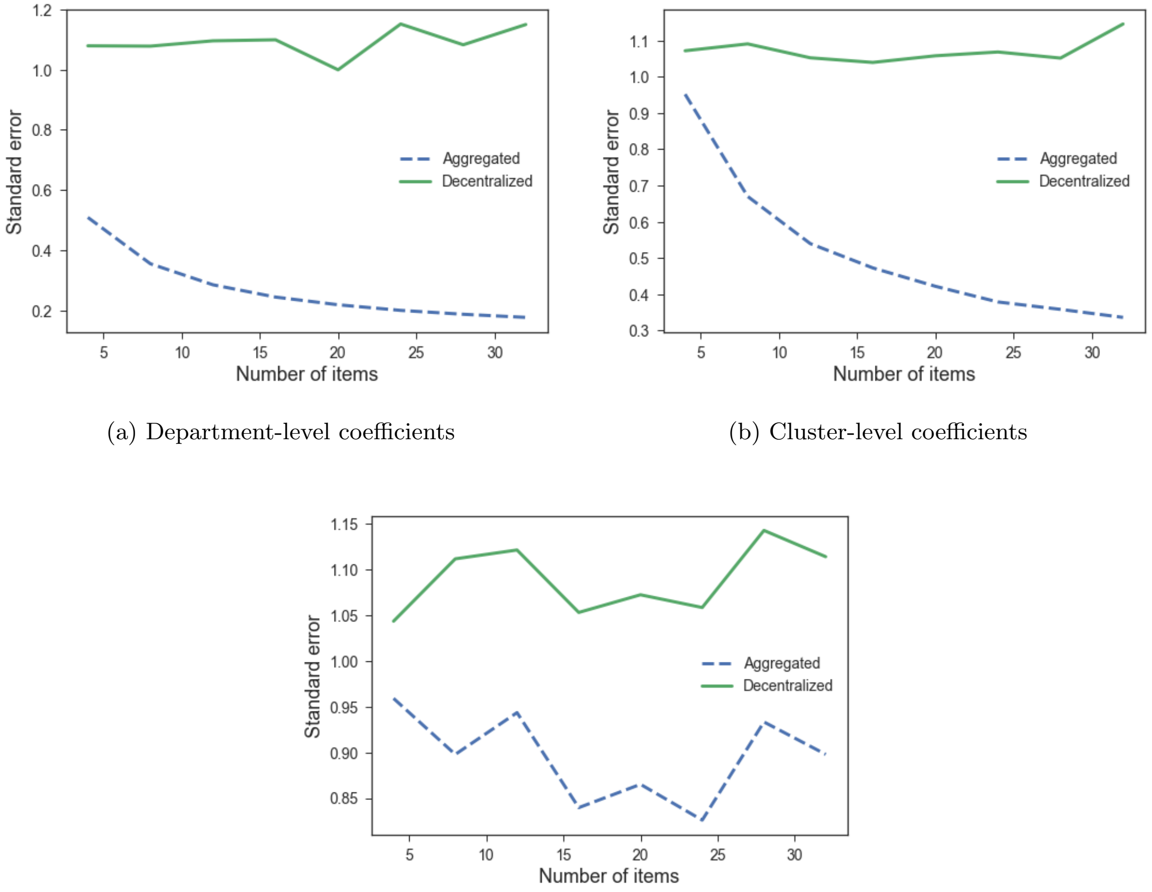
where $\xrightarrow{d}$ refers to convergence in distribution.

(b) If $\mathcal{I}_i(\beta_i)$ is diagonal for all i (e.g., in the linear regression model where different columns of X_i are orthogonal with each other for each i), then $\mathcal{I}(\beta)$ is also diagonal. In this case, there exists a constant $\kappa_{i,l} > 0$ for any (i, l), such that

$$\lim_{m \to +\infty} m \cdot n_{i,l} \cdot \mathit{Var}(\hat{\beta}_{i,l}) = \kappa_{i,l}, \text{ for } i = 1, 2, \ldots, n, \ l = 1, 2, \ldots, d,$$
$$\lim_{m \to +\infty} m \cdot \mathit{Var}(\hat{b}_{i,l}) = \kappa_{i,l}, \text{ for } i = 1, 2, \ldots, n, \ l = 1, 2, \ldots, d.$$

Since $n_{i,l} > 1$ for $l \in D_s \cup D_c$, the aggregated estimation yields a smaller asymptotic variance relative to the decentralized estimation for the coefficients of features at aggregated and cluster levels.

When the number of observations m becomes larger, the variance of the estimated coefficients of both models will shrink to zero. What makes the aggregated model more powerful is its capability to pool the data from different items, thus further reducing the variance of estimation, as shown in Proposition 6. In particular, if a feature is at the aggregated or cluster level, the aggregated model will use at least twice as many observations as the decentralized model to estimate the coefficient of this feature. Hence, the variance will shrink faster especially when n is large. In practice, a typical retail department consists of a large number of items ($n > 100$), so that the aggregated model can be much more efficient than the decentralized model. Proposition 6 further shows that, for the ordinary least squares (OLS) setting, the variances of aggregated and decentralized models can be computed in closed form and, thus, directly comparable. For a general non-linear GLM setting (e.g., logistic regression), however, a closed-form expression of variances cannot be derived. Instead, we will convey the efficiency improvement of the aggregated estimator relative to the decentralized estimator by numerically computing the standard error for the estimated coefficients at different aggregation levels.



(a) Department-level coefficients

(b) Cluster-level coefficients

(c) SKU-level coefficients

Figure 3.1: Comparison of standard errors for the aggregated and decentralized models

In Figure 3.1, we consider a simple illustrative example where we fix the number of observations for each item ($m = 50$) and the number of clusters ($k = 4$).[2] For each value of n, we generate 100 independent instances to compute the average standard error of the estimated coefficients for both the aggregated and decentralized estimators (for each type of feature). As n increases, the standard errors of the estimated coefficients for department and cluster levels decrease monotonically for the aggregated model. In contrast, n does not affect the standard errors for the decentralized model. The estimated coefficients improve substantially when n increases. These plots demonstrate the significant efficiency improvement (i.e., reducing variance) of the aggregated model relative to the decentralized model when there are department- and cluster-level features. We next derive probabilistic bounds for the mean squared error under the OLS setting for both the decentralized and aggregated models.

[2] Parameters for Figure 3.1: $d = 3$ (one feature at each level), β is obtained from a uniform $[-2, 2]$, X from a uniform $[0, 1]$, and $\sigma^2 = 0.1$ (for more details, see Section 3.5). For each n, we generate the data and estimate both models.

Proposition 7. *Under the OLS setting, we have*

$$\mathbb{P} \quad \frac{||X\beta - X\hat{\beta}||_2^2}{n \times m} \leq \frac{\sigma^2 \left(2\sqrt{\gamma d_x} + 2\gamma + d_x\right)}{n \times m} \quad \geq 1 - \exp(-\gamma)$$

for the decentralized model, and

$$\mathbb{P} \left(\frac{{}^{n}_{i=1} ||X_i b_i - X_i \hat{b}_i||_2^2}{n \times m} \leq \frac{\sigma^2 \quad 2\sqrt{\gamma\,(nd)} + 2\gamma + nd}{n \times m} \right) \geq 1 - \exp(-\gamma)$$

for the aggregated model. Furthermore, when

$$d_x \leq nd - 2 \quad \sqrt{\gamma nd} + \sqrt{\gamma nd - 2\gamma\sqrt{nd} - \gamma^2}$$

there exists a threshold value Γ *such that,*

$$\mathbb{P} \quad \frac{||X\beta - X\hat{\beta}||_2^2}{n \times m} \leq \Gamma \textit{ and } \Gamma \leq \frac{{}^{n}_{i=1} ||X_i b_i - X_i \hat{b}_i||_2^2}{n \times m} \quad \geq 1 - \exp(-\gamma), \tag{3.3}$$

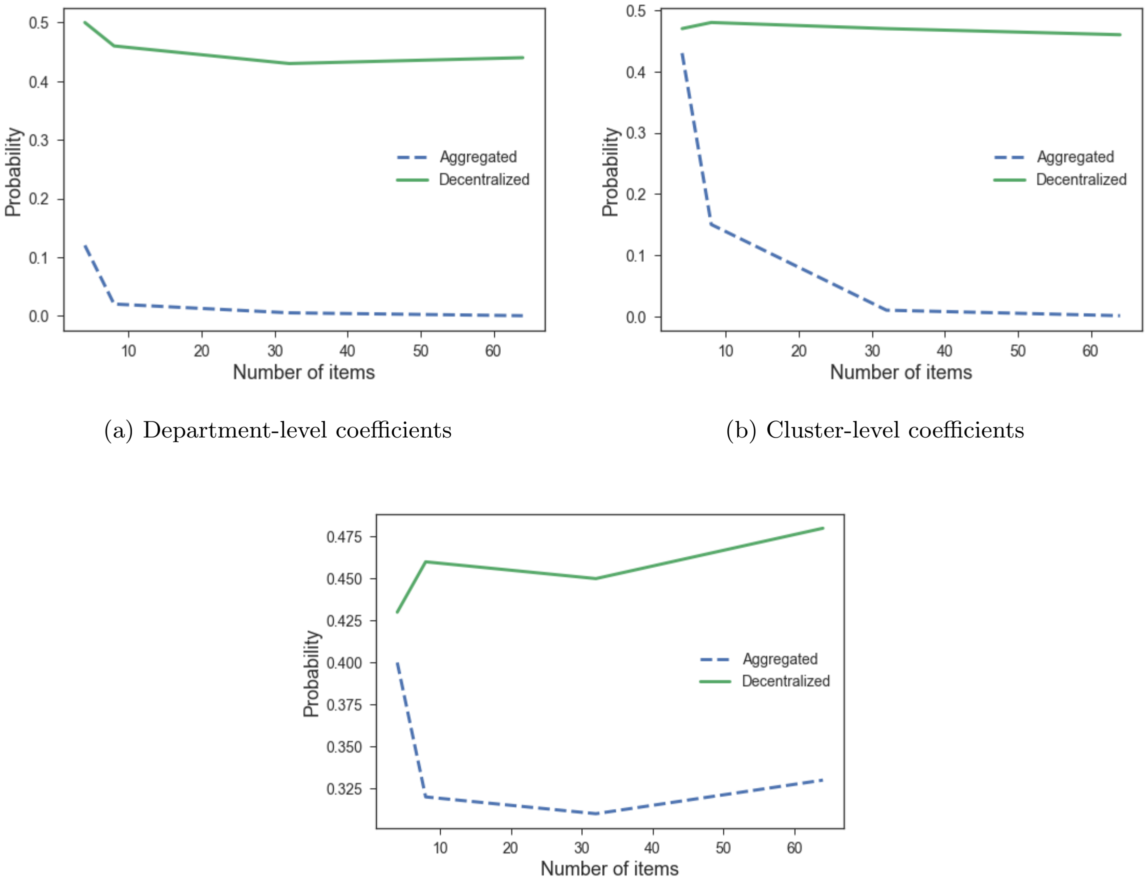
which implies that the aggregated model outperforms the decentralized model with high probability (if we set γ *large enough).*

Proposition 7 implies that, with high probability, the estimation error of the aggregated model is smaller relative to the decentralized model as long as the number of coefficients is small. Note that condition (3.3) is well defined only when $\gamma \leq nd - 2\sqrt{nd}$. Table 3.1 illustrates the result of Proposition 7, that is, how large d_x can be relative to the total number of features nd.

Table 3.1: Maximum value of d_x

γ	Probability	nd	d_x^*
1	0.632	500	412
2	0.865	1,000	824
5	0.993	3,000	2,514
10	0.999	5,000	4,112

As we can see from Table 3.1, the upper bound on d_x (denoted by d_x^*) is relatively easy to satisfy. For example, when $nd = 1,000$, it means that for the decentralized model one needs to estimate 1,000 parameters. In this case, the aggregated model will outperform the decentralized model with probability (at least) 0.865 as long as the total number of features is less than 824. Suppose now that the number of parameters in the original model is large (e.g., $nd = 5,000$). As long as there is a non-negligible number of department- and cluster-level features to reduce the number of coefficients to 4,112, the aggregated model will outperform the decentralized model with probability very close to 1. This ultimately illustrates the power of aggregating data and reducing dimensionality to improve prediction accuracy.



(a) Department-level coefficients

(b) Cluster-level coefficients

(c) SKU-level coefficients

Figure 3.2: Comparison of large estimation error probability for aggregated and decentralized models

For a non-linear GLM model (e.g., logistic regression), since the prediction accuracy does not have a closed form expression, we study the probabilistic bound computationally. For a given coefficient, we compute $\mathbb{P}(|\hat{\beta}_{i,l} - \beta_{i,l}| > \eta)$, which measures the confidence level of the estimate. As we can see from Figure 3.2, regardless of the aggregation level, the probability that the estimated coefficient is far from the true value is lower for the aggregated model relative to the decentralized model, especially when n is large.[3]

To conclude this section, we remark that all our analysis has focused on comparing the aggregated and decentralized models. We note that the data aggregation level and cluster structure are not known a priori, but are identified via the hypothesis testing and the k-means steps of the DAC. As a result, one could expect additional biases and increased variances for the DAC relative to the aggregated model. Ultimately, one may question whether the value of data aggregation and clustering remains significant for the DAC. Our simulation and real data studies (see Sections 3.5 and 3.6) clearly convey that our proposed DAC algorithm efficiently

[3] Parameters for Figure 3.2: $m = 200$, $k = 4$, $d = 3$ (one feature at each level), β is obtained from a uniform $[-2, 2]$, X from a uniform $[0, 1]$, and $\eta = 0.4$. For each n, we generate the data and fit both models. Since we generate 100 independent instances for each n, we count the number of instances where $|\hat{\beta} - \beta| > \eta$ to compute the probability.

identifies and leverages the data aggregation and cluster structure, and hence substantially improves the out-of-sample prediction accuracy relative to several benchmarks.

3.5 Simulated Experiments

In this section, we conduct computational experiments using simulated data. We focus on the predictive power of our method and illustrate the improvement in prediction accuracy relative to several benchmarks. We consider two settings under the GLM framework: OLS and logistic regression. The model performance is evaluated using the out-of-sample R^2 for OLS and the area under the curve (AUC) score for logistic regression. We also undertake a comprehensive sensitivity analysis to examine how the different parameters affect the model performance.

3.5.1 Linear Regression

The data is assumed to be generated from the following linear model:

$$Y_{i,j} = \underset{l\in D_s}{} X^l_{i,j}\beta^s_l + \underset{l\in D_n}{} X^l_{i,j}\beta^n_{i,l} + \underset{l\in D_c}{} X^l_{i,j}\beta^c_{\mathcal{C}(i),l} + \epsilon_{i,j}, \qquad i = 1,\ldots,n \text{ and } j = 1,\ldots,m,$$

where $\epsilon_{i,j} \sim N(0,\sigma^2)$ are independent and identically distributed random variables. Each data point, $X^l_{i,j}$, is generated randomly from a uniform $[0,1]$ distribution, and each β coefficient is obtained from a uniform $[-2,2]$ distribution. We fix the number of clusters $k = 5$ and vary the parameters $\{n,d,m,\sigma^2,p,q\}$ one at a time. The definition and range of values for these parameters are reported in Table 3.2. The parameters p and q represent the probability that a given feature is modeled at the department and cluster levels, respectively (different features are drawn independently).

Table 3.2: Parameters used in Section 3.5.1

Parameter	Range of values
Number of items (n)	$[10, 150]$
Number of features (d)	$[2, 15]$
Number of observations (m)	$[10, 100]$
Variance of the noise (σ^2)	$[0.05, 0.25]$
Department-level probability (p)	$[0, 2/3]$ or $[0, 1]$
Cluster-level probability (q)	$[0, 2/3]$ or $[0, 1]$

It is important to note that the DAC implementation admits three design parameters: θ, R_U, and R_L in addition to the number of clusters k. These three parameters represent the strictness of our algorithm in determining whether or not a feature should be aggregated. Specifically, θ is the p-value cut-off for statistical significance and is usually 0.05 or 0.01. The parameters R_U and R_L represent the thresholds for the ratio of non-rejected hypotheses. For example, suppose that the percentage of non-rejected hypotheses for feature j is $R_j = 0.3$ (i.e., 30% of the items have statistically close estimated coefficients). Then, we label feature j as a department-level feature if $R_j > R_U$ and a SKU-level feature if $R_j < R_L$. For any intermediate value $R_j \in [R_L, R_U]$, we will label feature j as a cluster-level feature. For instance, one can set $R_L = 0.1$ and $R_U = 0.9$. The parameters θ, R_U, and R_L provide us with flexibility in the tolerance level of the algorithm. When using real data (see Section 3.6), we will set their values using cross-validation.

To test the performance of our algorithm, we consider the four following benchmarks: Decentralized, Decentralized-Lasso, Centralized, and Clustering. For each instance (i.e., a specific combination of $\{n, d, m, \sigma, p, q\}$), we generate 100 independent trials (datasets) and use 70% as training and 30% as testing. We then report the average out-of-sample R^2. Below is a description of all the methods we consider:

1. **DAC**: We implement our algorithm with $\theta = 0.01$, $R_U = 0.9$, and $R_L = 0.1$.

2. **Decentralized**: We estimate a simple OLS model for each item separately (i.e., n models).

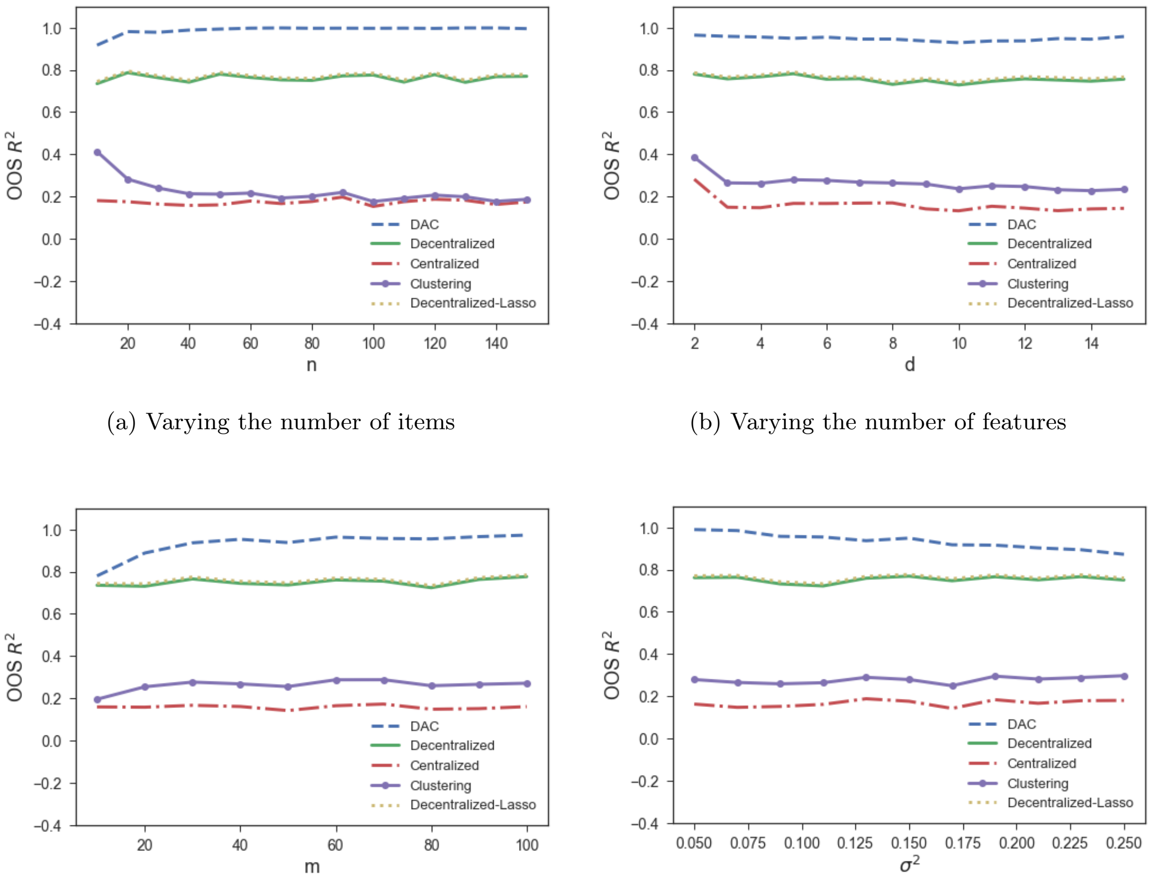
3. **Decentralized-Lasso**: Same as the decentralized method while adding a L_1 regularization term to each OLS model.

4. **Centralized**: This is a naïve OLS model where we assume that, for each feature, all the items have the same coefficient.

5. **Clustering**: We first cluster the items using k-means based on the mean values of the features. We then fit an OLS model for each cluster.

As we can see from Figure 3.3, our algorithm outperforms all the benchmarks in all settings in terms of out-of-sample R^2. As we increase the number of items or the number of observations, the prediction accuracy of DAC quickly converges to 1, as opposed to the other methods. This clearly demonstrates the power of data aggregation and cluster structure of our algorithm. As expected, a higher σ^2 has a negative impact on the prediction accuracy as it makes structure identification more challenging. Still, our proposed algorithm depicts a substantial advantage relative to the four benchmarks. Finally, varying the number of features does not affect the performance of any of the methods (the performance of each method depends on the proportion of the different feature types and not on the absolute number of features). In addition, we can observe that the DAC has a smoother curve (i.e., fewer "bumps") than the other methods. This implies that our method generates a more stable prediction and, in general, has a smaller variance.

Figure 3.4 presents the performance of the methods as we vary the structure probability of the features in terms of aggregation level. The first two plots show that if a large proportion of the features are at the department-level (i.e., p is close to 1), all five methods perform well (the DAC still performs best in all cases). However, for instances where the structure is more diverse, our algorithm significantly outperforms the four benchmarks. The bottom panels in Figure 3.4 convey a similar message, except that the Clustering and Centralized methods have a poor performance when the number of department-level features is small.



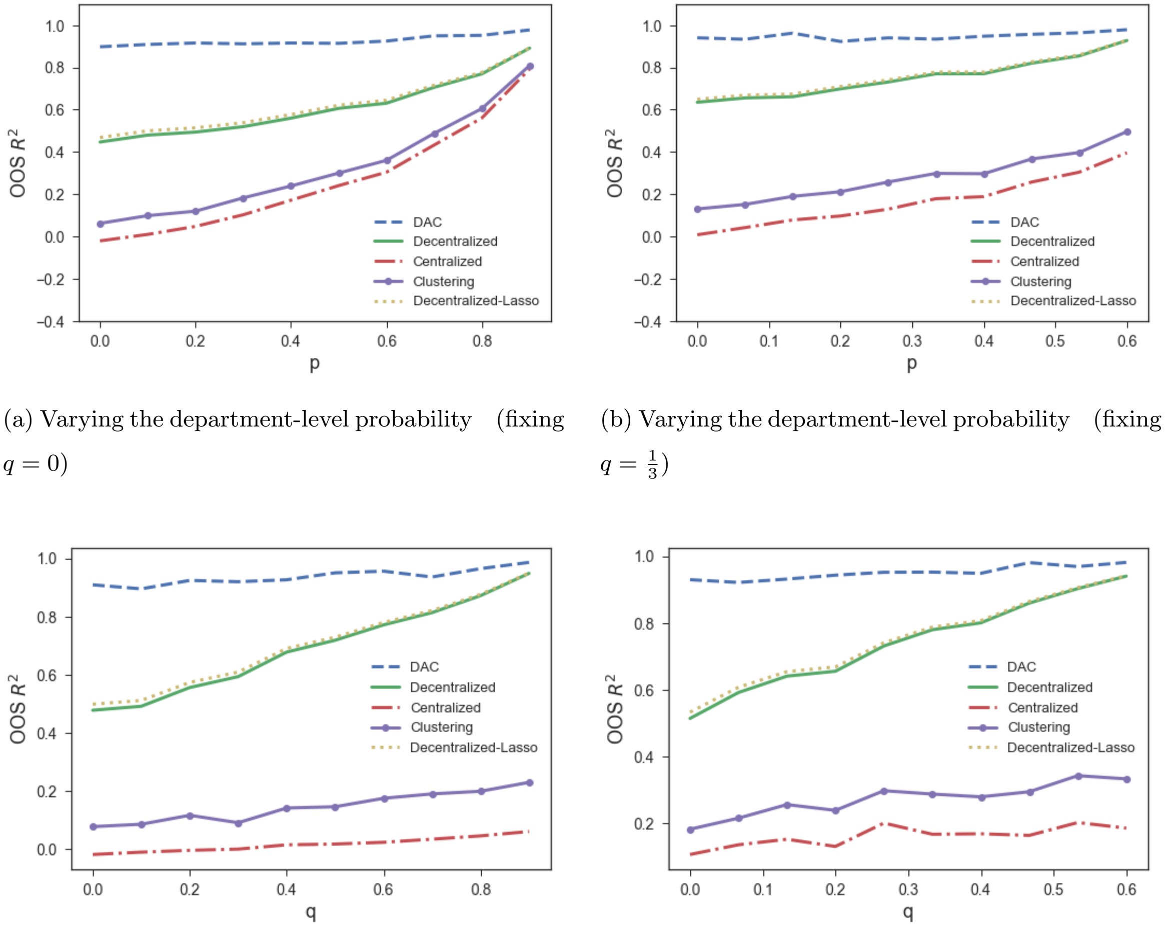
(a) Varying the number of items

(b) Varying the number of features

(c) Varying the number of observations

(d) Varying the noise magnitude

Figure 3.3: Performance comparison of different prediction models (for linear regression)



(a) Varying the department-level probability (fixing $q = 0$)

(b) Varying the department-level probability (fixing $q = \frac{1}{3}$)

(c) Varying the cluster-level probability (fixing $p = 0$)

(d) Varying the cluster-level probability (fixing $p = \frac{1}{3}$)

Figure 3.4: Performance comparison of different prediction models (for linear regression)

3.5.2 Logistic Regression

In this section, we present computational experiments for a classification problem in which the data generating process is the logistic regression model, that is,

$$Y_{i,j} \sim Bernoulli\,(\mu_{i,j})\,, \qquad i = 1,\ldots,n \text{ and } j = 1,\ldots,m,$$

$$\mu_{i,j} = \text{logit} \quad \underset{l\in D_s}{} X^l_{i,j}\beta^s_l + \underset{l\in D_n}{} X^l_{i,j}\beta^n_{i,l} + \underset{l\in D_c}{} X^l_{i,j}\beta^c_{\mathcal{C}(i),l} \quad ,$$

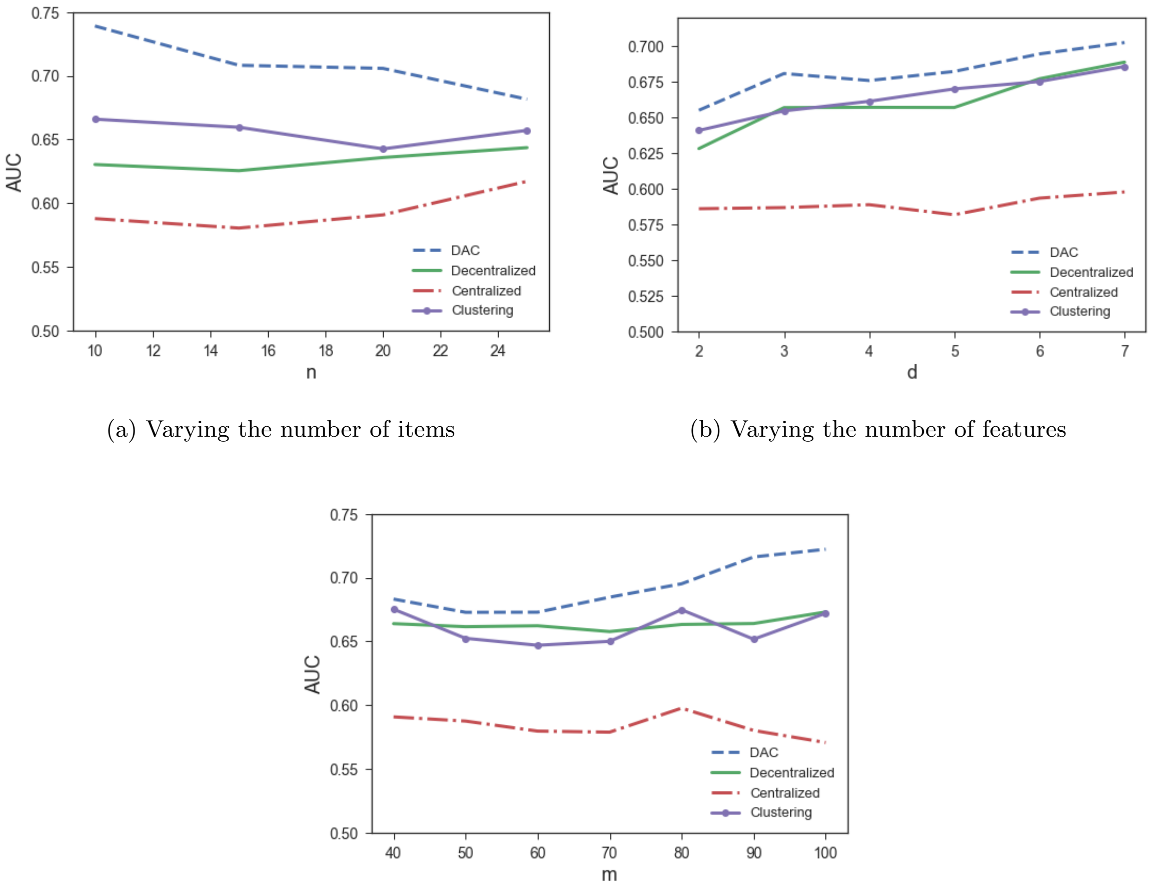
where logit(z) := $\frac{\exp(z)}{1+\exp(z)}$. We use a similar setting as in Section 3.5.1. We first generate the data matrix X using a uniform $[0,1]$ distribution and the β coefficients from a uniform $[-2,2]$ distribution. The outcome variable Y is then generated based on a Bernoulli distribution with parameter $\mu = \text{logit}(X\beta)$. As in the OLS case, we vary one parameter at a time. The ranges of the parameters are summarized in Table 3.3.

Table 3.3: Parameters used in Section 3.5.2

Parameter	Range of values
Number of items (n)	$[10,25]$
Number of features (d)	$[2,7]$
Number of observations (m)	$[40,100]$
Department-level probability (p)	$[0,2/3]$ or $[0,1]$
Cluster-level probability (q)	$[0,2/3]$ or $[0,1]$

Following several prior studies on binary classification problems, we use the AUC as the metric to evaluate the performance of the different models. AUC is defined as the area under the receiver operating characteristic (ROC) curve (see, e.g., [12]). It can be interpreted as the probability that a prediction model is correctly ranking a random positive outcome higher than a random negative outcome. We compare our algorithm relative to three benchmarks: Decentralized, Centralized, and Clustering (definitions are similar to Section 3.5.1).[4] For each instance, we generate 100 independent trials and report the average out-of-sample AUC scores.

[4] Since estimating a decentralized model with L_1 regularization is computationally prohibitive for the logistic regression setting, we only show the performance of the decentralized model without regularization.



(a) Varying the number of items

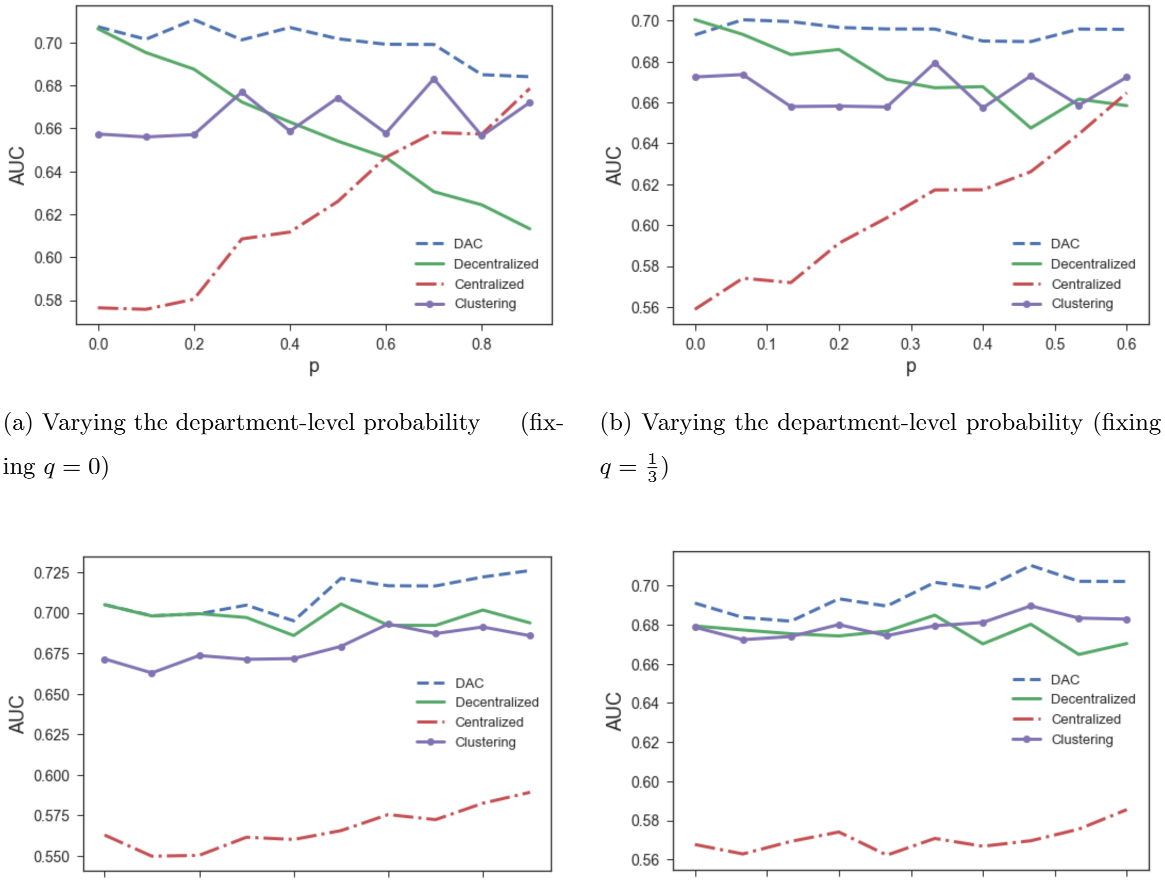
(b) Varying the number of features

(c) Varying the number of observations

Figure 3.5: Performance comparison of different prediction models (for logistic regression)

As we can see from Figures 3.5 and 3.6, our method outperforms the benchmarks in all cases. Regardless of how we vary $\{n, m, d\}$, the DAC outperforms the three other methods in terms of prediction accuracy. As expected, if the number of department- or cluster- level features are low (i.e., p or q are small), the advantage of the DAC is reduced. However, when at least 30% of the features are at the department or cluster level, our method significantly outperforms the three benchmarks.

To summarize, our simulation studies exhibit a substantial and robust performance improvement for our proposed algorithm relative to several benchmarks (which are commonly used in practice and in the literature). For both the regression and classification problems, the DAC efficiently aggregates the data and identifies the cluster structure of the items, and thus improves the prediction accuracy. In the next section, we apply the DAC to actual retail data to showcase the benefits in a practical business setting.



(a) Varying the department-level probability (fixing $q = 0$)

(b) Varying the department-level probability (fixing $q = \frac{1}{3}$)

(c) Varying the cluster-level probability (fixing $p = 0$)

(d) Varying the cluster-level probability (fixing $p = \frac{1}{3}$)

Figure 3.6: Performance comparison of different prediction models (for logistic regression)

3.6 Applying DAC to Retail Data

In this section, we apply the DAC using a retail dataset from a large global retail firm (we do not reveal the name of the retailer due to a non-disclosure agreement). We first provide a detailed description of the data, and then test the prediction performance of our model relative to several benchmarks. Finally, based on our computational findings, we draw useful managerial insights that can help retailers infer which features should be aggregated in practice.

3.6.1 Data

We have access to the online sales data from the retailer. The dataset is comprehensive and includes several departments. Specifically, the data records weekly sales information of five departments between November 2013 and October 2016. A typical department comprises of 80-150 SKUs. In addition to the weekly sales information, the dataset also includes weekly price, promotion indicator (whether or not an item was promoted), vendor, and the color of the SKU. Table 3.4 summarizes the specifics of each department. The size corresponds to the number of items in each department and the number in parenthesis are the standard deviations.

Table 3.4: Summary statistics of each department

Dept	**Size**	**Observations**	**Weekly sales**	**Price**	**Promo Frequency**	**Discount rate**
1	117	19,064	108.11 (377.45)	42.92 (38.80)	31.3%	4.2%
2	134	20,826	254.23 (517.07)	8.94 (8.67)	7.7%	6.4%
3	113	17,207	200.17 (452.97)	12.63 (7.02)	8.9%	37.7%
4	125	14,457	68.68 (391.52)	97.61 (67.12)	8.5%	1.5%
5	84	12,597	115.62 (337.29)	22.17 (17.37)	6.3%	6.7%

As we can see from Table 3.4, each department has a large number of observations (observations are at the week-SKU level). In addition, we have a great variation in terms of weekly sales, price, promotion frequency, and discount rate across the different departments. Table 3.5 provides a brief description of the different fields in our dataset. The effective weekly price is computed as the total weekly revenue divided by the total weekly sales. Functionality is a segmentation hierarchy used by the firm to classify several SKUs from the same department into sub-categories.

Table 3.5: Fields in our dataset (observations are aggregated at the week-SKU level)

Fields	**Description**
SKU ID	Unique SKU ID
Week	Week index
Year	Year index
Units	Total weekly sales of a specific SKU
Price	Effective weekly price of a specific SKU
PromoFlag	Whether there was a promotion during that week
Functionality	Class index of a specific SKU
Color	Color of a specific SKU
Vendor	Vendor of a specific SKU

Based on the features available in our data, we estimate the following baseline regression model:

$$\begin{aligned} Y_{i,t} &= \beta^i_{\text{Trend}} \cdot T_{i,t} + \beta^i_0 \cdot p_{i,t} + \beta^i_1 \cdot \text{PromoFlag}_{i,t} + \beta^i_2 \cdot \text{Fatigue}_{i,t} + \beta^i_3 \cdot \text{Seasonality}_{i,t} + \\ &+ \beta^i_4 \cdot \text{Functionality}_{i,t} + \beta^i_5 \cdot \text{Color}_{i,t} + \beta^i_6 \cdot \text{Vendor}_{i,t} + \epsilon_{i,t}, \end{aligned} \tag{3.4}$$

Equation (3.4) includes the following features:

1. $Y_{i,t}$: total weekly sales of item i in week t (our dependent variable),
2. $T_{i,t}$: trend variable of item i. We normalize the year so that $T_i = 0, 1, 2, 3$,
3. $p_{i,t}$: effective price of item i in week t,
4. $\text{PromoFlag}_{i,t}$: binary variable indicating whether there is a promotion for item i in week t,
5. $\text{Fatigue}_{i,t}$: number of weeks since the most recent promotion for item i. If there is no previous promotion, $Fatigue_{i,t} = 0$. This feature allows us to capture the post-promotion dip effect.
6. Seasonality: categorical variable that measures the weekly or monthly effect on sales. We use the one-hot encoding in our model.[5]
7. $\epsilon_{i,t}$: additive unobserved error.

The remaining three variables are categorical variables indicating the web class index (Functionality), Color, and Vendor of the SKU, respectively.

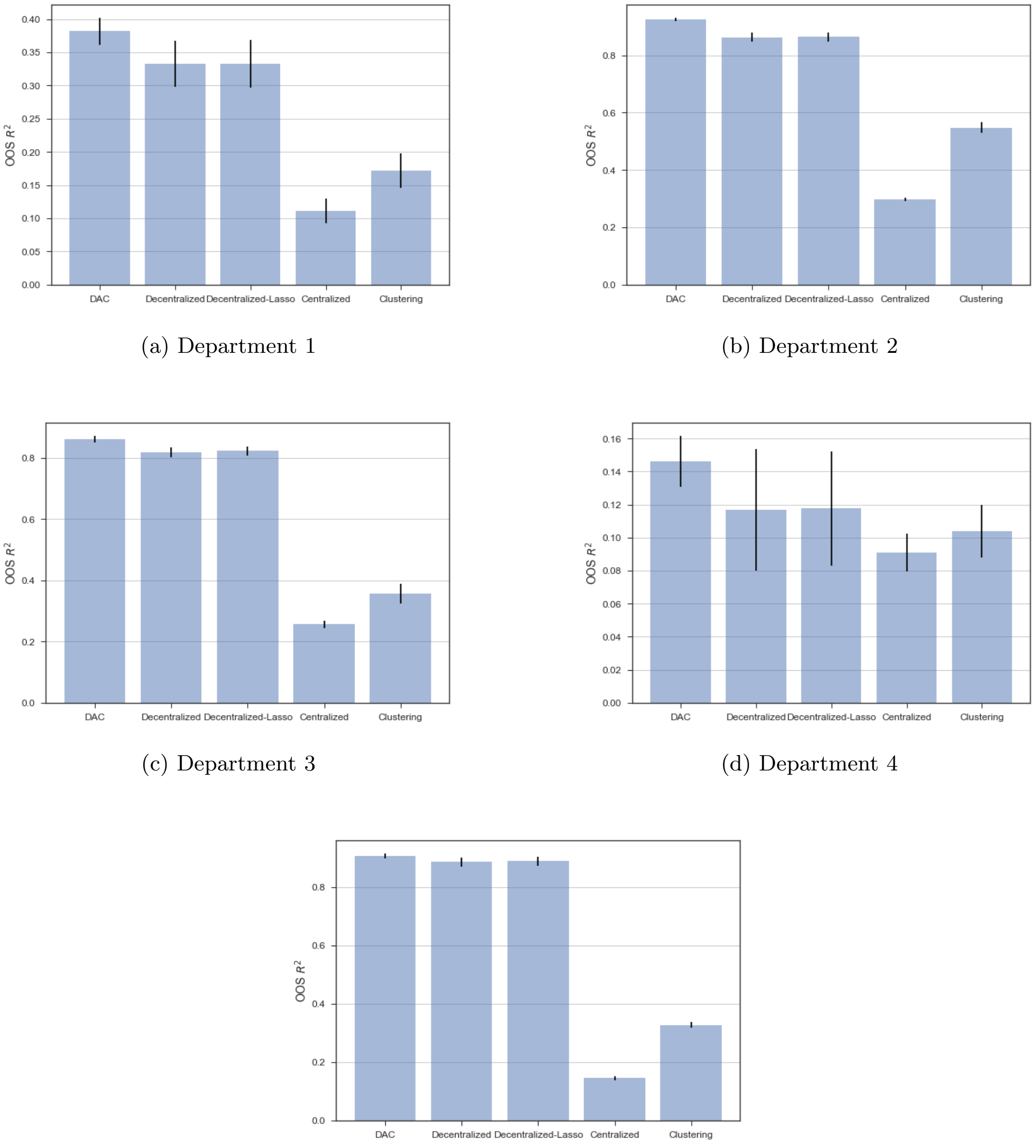
3.6.2 Prediction Performance

As in Section 3.5, we investigate the performance of our algorithm relative to the same four benchmarks (Decentralized, Decentralized-Lasso, Centralized, and Clustering). It is important to mention that when

[5] https://en.wikipedia.org/wiki/One-hot

implementing our algorithm using real data, we need to slightly adapt the clustering step. To make sure that we output a single cluster structure, we first collect the estimated coefficients from all cluster-level features, and then fit a multi-dimensional k-means model. To avoid overfitting, we also include a L_1-regularization term in the last step of the aggregated estimation.

Since DAC(k, θ, R_U, R_L) has four design parameters, we use cross-validation for model selection. For each department, we first randomly split the data into training (70%) and testing (30%). We assume that each design parameter lies within a pre-specified range: $k \in \{3, 4, \ldots, 10\}$, $\theta \in \{0.01, 0.05, 0.1\}$, $R_U \in \{0.7, 0.8, 0.9\}$, and $R_L \in \{0.1, 0.2, 0.3\}$. For each combination of design parameters, we perform a five-fold cross-validation by fitting the model on 80% of the training data and compute the R^2 based on the remaining 20% of the data. This procedure is repeated five times for each different parameter combination, and we compute the average R^2 over the five folds. We next select the best model based on the average cross-validation performance. Finally, the out-of-sample R^2 is computed using the test set. Furthermore, since the train/cross-validation/test split is done randomly, we conduct 100 independent trials and report both the mean and standard deviation of the out-of-sample R^2. The following bar charts summarize the prediction performance of the DAC.



(a) Department 1

(b) Department 2

(c) Department 3

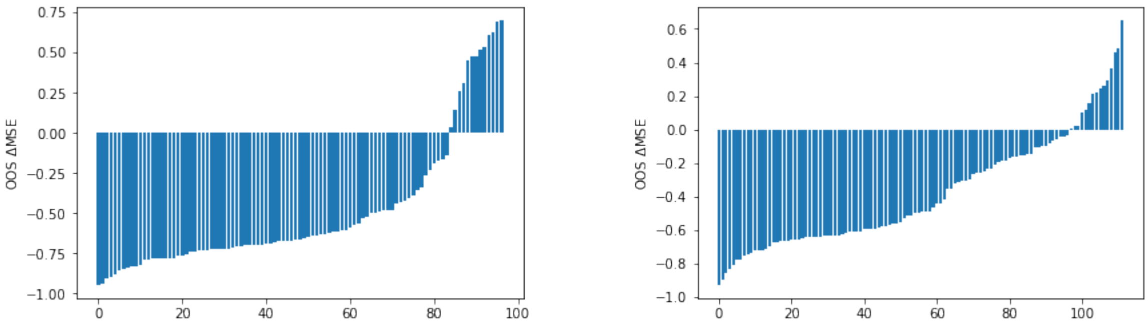
(d) Department 4

(e) Department 5

Figure 3.7: Performance comparison using real data

In Figure 3.7, each bar represents the average out-of-sample performance and the length of the vertical line corresponds to the standard deviation across 100 independent trials. As we can see, for all five departments, our algorithm not only achieves a better average prediction performance but also has a smaller variance. This shows that our algorithm is robust to different train/test splits, which is very desirable in practice. In addition, our method outperforms all four benchmarks, regardless of data quality. More precisely, Department 2 seems to have a high-quality data, whereas the data for Department 4 seems to be of lower quality (the number of observations per SKU is the lowest for Department 4 and the variability is high). Irrespective of the data quality, the DAC yields a clear improvement in prediction accuracy.

We next present a per-item comparison between Departments 1 and 2. Specifically, we compare the performance of our algorithm relative to the second best method, that is, Decentralized-Lasso for each item in the department. To mitigate the effect of outliers, we remove the bottom and top 5% SKUs in terms of mean squared error (MSE) for both methods.



(a) Department 1

(b) Department 2

Figure 3.8: Per-item prediction accuracy comparison

The y-axis in Figure 3.8 is the relative out-of-sample MSE, that is, $\frac{MSE_i(DAC)-MSE_i(Dece)}{MSE_i(Dece)}$, which measures the percentage improvement of DAC relative to Decentralized-Lasso for each item i in the department. Note that we report the per-item MSE instead of the per-item R^2, given that the per-item R^2 is not a well-defined metric. Our algorithm improves the demand prediction for more than 70% of the items in both departments. More generally, when applying the DAC to all five departments, we obtain an improvement for 412 out of the 573 SKUs (i.e., 71.9% of the items). Even though the Decentralized-Lasso method yields a good performance in many cases (recall that it avoids overfitting), its performance can be limited when the number of observations per item is small and the number of features is large. On the other hand, our algorithm can adaptively identify the right level of aggregation for each feature, thus alleviating the over-fitting issue and increasing prediction accuracy.

3.6.3 Managerial Insights

So far, we focused on the prediction performance of the DAC. We next apply the DAC to our dataset and examine the estimation output. Our goal is to draw managerial insights on the hierarchical structure of the features among the different SKUs. We next summarize our findings.

- The DAC can significantly reduce the model dimension. In Table 3.6, we report the number of estimated coefficients for the Decentralized and DAC methods across all five departments. For Departments 1 and 2, the number of estimated coefficients reduces by 40%, and for Departments 3–5, the reduction exceeds 50%. The results in Table 3.6 confirm that shared coefficients do occur in practice, and that dynamic data aggregation can play an important role in correctly identifying the aggregation structure.

Table 3.6: Number of estimated coefficients

Department	**Decentralized**	**DAC**
1	7,488	4,360
2	6,298	3,912
3	2,712	1,080
4	3,000	892
5	1,848	729

- Practitioners often argue that seasonality features should be aggregated at the department level for demand prediction (e.g., [25, 106]). Using our retail dataset, we discover that this is indeed the case. If we model the seasonality at the month level (i.e., we use 12 dummy variables for each calendar month), we find that at least 10 out of the 12 variables should be estimated at the department level, for each of the five departments. If we instead model the seasonality at the week level (i.e., we use 52 dummy variables for each week of the year), we find that over 90% of the variables should be estimated at the department level. Thus, our findings validate and refine a well-known business practice.

- The price feature is unarguably one of the most important features for demand prediction in retail. According to our estimation results, for departments with a heterogeneous item collection (Departments 1, 2, and 4), we obtain a distinct coefficient for the price feature, implying that the price coefficient should be estimated at the SKU level. On the other hand, for departments in which the product discrepancy is low (Departments 3 and 5), the DAC infers that the price coefficient should be estimated at the cluster level.

- We find that the fatigue and promotion features should be estimated at the department level, for all five departments. This is an interesting insight that can guide retailers when deciding their promotion strategy.

- We obtain that all vendor and color dummy variables should be estimated at the SKU level. This not surprising given that most vendor-color combinations are unique for a specific SKU.

- The functionality dummy variables have different aggregation levels and cluster structures. Interestingly, most cluster-level features come from this variable. One possible explanation is that the functionality feature is obtained based on the hierarchy structure used by the company. Thus, SKUs with similar characteristics are usually labeled under the same functionality, making the cluster structure more prominent for functionality features. Retailers can use such results to potentially revise and improve their hierarchical structure and product segmentation.

3.7 Conclusion

Demand prediction (or sales forecast) is an important task faced by most retailers. Improving the prediction accuracy and drawing insights on data aggregation can significantly impact retailers' decisions and profits. When designing and estimating predictive models, retailers need to decide the aggregation level of each model feature. Some features may be estimated at the SKU level, others at the department level, and the rest at a cluster level. Traditionally, this problem was addressed by trial-and-error or by relying on past experience. It is common to see data scientists testing a multitude of model specifications until they find the best aggregation level for each feature. Such an approach can be tedious and does not scale for models with a large number of features. The goal of this chapter is to develop an efficient method to simultaneously determine (i) the right aggregation level of each feature, (ii) the underlying cluster structure, and (iii) the estimated coefficients.

We propose a method referred to as the Dynamic Aggregation with Clustering (DAC) algorithm. The DAC can dynamically determine the right aggregation level and identify the cluster structure of the items. This method is tractable even when the data dimensionality is high and can significantly improve the efficiency in estimating the model coefficients. We first derive several analytical results to demonstrate the benefit of aggregating similar coefficients. Specifically, we show that the DAC yields a consistent estimate along with improved asymptotic properties relative to the traditional OLS method. We then go beyond the theory and implement the DAC using a large retail dataset. In all our computational tests, we observe that the DAC significantly improves the prediction accuracy relative to several benchmarks. Finally, we convey that our method can help retailers discover useful insights from their data.

3.8 Appendix

3.8.1 Two Potential Methods

In this section, we elaborate on the key difficulties of two intuitive methods—*constrained MLE* and *iterative optimization*—in estimating Equation (3.1).

Constrained MLE

Constrained MLE adapts the standard MLE approach to account for the equality constraints $\beta_{i,l} = \beta_{j,l}$, for all i, j if l is an aggregated-level feature; and $\beta_{i,l} = \beta_{j,l}$, for all i, j in the same cluster if l is a cluster-level feature. This method, however, is challenging to implement. We next use a simple example to illustrate the main difficulty. Consider a linear regression model with two items (each with one feature) but the level of aggregation is unknown. The data matrix can be written as follows:

$$X = \begin{bmatrix} X_1^1 & 0 & X_1^1 \\ X_2^1 & 0 & X_2^1 \\ ... & ... & ... \\ X_m^1 & 0 & X_m^1 \\ 0 & X_1^2 & X_1^2 \\ 0 & X_2^2 & X_2^2 \\ ... & ... & ... \\ 0 & X_m^2 & X_m^2 \end{bmatrix}.$$

The data matrix X enumerates all possible column combinations (three in this case), which represent all possible options for feature aggregation. If the feature is at the SKU level, then only the first two columns will have non-zero coefficients. On the other hand, if the feature is at the department level, only the last column will have a non-zero coefficient. We denote the vector of parameters as $\beta = (\beta_1, \beta_2, \beta_{12})$. The aggregation level inference problem can be formulated as the following constrained MLE problem (for linear regression, MLE is equivalent to minimizing the squared error):

$$\begin{aligned} \min_{\beta} \quad & ||X\beta - y||^2 \\ \text{s.t.} \quad & \beta_1\beta_{12} = 0, \\ & \beta_2\beta_{12} = 0. \end{aligned}$$

Applying the KKT condition, we obtain the following matrix-form optimality condition:

$$2X^TX\beta - 2X^Ty + B^T\lambda = 0,$$

where

$$B = \begin{bmatrix} \beta_{12} & 0 & \beta_1 \\ 0 & \beta_{12} & \beta_2 \end{bmatrix}.$$

The complete KKT condition is then given by:

$$\begin{bmatrix} 2X^TX & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \beta \\ \lambda \end{bmatrix} = \begin{bmatrix} 2X^Ty \\ 0 \end{bmatrix}$$

which is a non-convex quadratic equation. Thus, the traditional approach to solve a constrained linear regression does not apply in this case. In addition, due to the non-convexity, it is not easy to solve the above optimization problem when the number of items and/or the number of features become large. For a non-linear GLM model, the constrained MLE approach is even more challenging. As a result, the constrained MLE does not seem to be an efficient method to solve our problem.

Iterative Optimization

The iterative optimization approach was proposed by [6]. We use the same example as the constrained MLE (i.e., a linear model with SKU- and department-level features only), to illustrate the difficulty of applying the iterative optimization procedure. We formulate the estimation problem as the following optimization program:

$$\begin{aligned} \min_{\delta,\beta_n,\beta_0} \quad & \mathop{}\limits_{j=1}^{m} \mathop{}\limits_{i=1}^{n} \; Y_{i,j} - \mathop{}\limits_{l=1}^{d} \; X_{i,j}^{l}\beta_{i,l}\delta_l + X_{i,j}^{l}\beta_{0,l}(1-\delta_l) \quad {}^{2} \\ \text{s.t.} \quad & \beta_{i,l} \in \mathbb{R}^d \;\; i = 1,2,\ldots,n, \;\; l = 1,2,\ldots,d, \\ & \delta_l \in \{0,1\} \;\; l = 1,2,\ldots,d, \end{aligned} \tag{3.5}$$

where i is the observation index, j the item index, and l the feature index. $\beta_0 = (\beta_{0,1}, \beta_{0,2}, \ldots, \beta_{0,d})'$ (resp. $\beta_j = (\beta_{j,1}, \beta_{j,2}, \ldots, \beta_{j,d})'$) are the department-level (resp. SKU-level) coefficients for the features. This iterative optimization formulation is a quadratic mixed-integer program. Note that the model is simplified since it does not include cluster-level coefficients.

One can solve the optimization problem in (3.5) by using the following iterative algorithm:

1. Randomly initialize $\hat{\delta}^0 \in \{0,1\}^d$ and solve for $\hat{\beta}^0 = \{\hat{\beta}^0_{i,l} : i = 1,2,\ldots,l = 1,2,\ldots,d\}$ by fixing $\hat{\delta} = \hat{\delta}^0$.

2. Starting from iteration $t = 1$, first solve for $\hat{\delta}^t$ by fixing $\beta = \hat{\beta}^{t-1}$; then solve for $\hat{\beta}^t$ by fixing $\delta = \hat{\delta}^t$.

3. Terminate Step 2 when $\hat{\delta}^t = \hat{\delta}^{t-1}$. The output is $(\hat{\beta}^t, \hat{\delta}^t)$.

The difficulty of the iterative optimization procedure is that without any prior information on β, the generation of coefficient vector is random. Therefore, if the generated coefficients are far from their true values, the procedure can converge to a local optimal solution, and hence potentially yield a large estimation error. Furthermore, it is not clear how to incorporate the (unknown) cluster structure into the iterative optimization procedure.

3.8.2 Proofs of Statements

Proof of Lemma 4

The proof follows from a standard result in statistics stating that the maximum likelihood estimator (MLE) is consistent under some regularity conditions which are satisfied by a generalized linear model. See, for example, [36] and [82].

Proof of Proposition 1

We analyze each aggregation level separately. The results build on Lemma 4 stating that all $\hat{b}_{i,j}$ are consistent.

- Case 1: Feature l is at the SKU level.

  In this case, $b_{1,l} = b_{k,l}$ for all k in the decentralized model. As a result, based on [71], for any $\eta > 0$, we have

  $$\lim_{m \to \infty} p\text{-value} = \lim_{m \to \infty} Pr(|\hat{b}_{1,l} - \hat{b}_{k,l}| < \eta) = 0. \tag{3.6}$$

  Therefore, the p-value converges to 0 as $m \to \infty$. Alternatively if $l \notin D_n$, there exists an item k', such that

  $$\lim_{m \to \infty} p\text{-value} = \lim_{m \to \infty} \mathbb{P}(|\hat{b}_{1,l} - \hat{b}_{k\ ,l}| \geq \eta) = 1. \tag{3.7}$$

  Combining inequalities (3.6) and (3.7) imply that the probability that we misclassify feature l (in terms of whether or not feature l is at the SKU level) converges to 0 as $m \to \infty$, that is, the DAC consistently identifies whether or not feature l is at the SKU level.

- Case 2: Feature l is at the department level.

  In this case, $b_{1,l} = b_{k,l}$ in the decentralized model. Again according to [71], for any $\eta > 0$, we have,

  $$\lim_{m \to \infty} p\text{-value} = \lim_{m \to \infty} Pr(|\hat{b}_{1,j} - \hat{b}_{k,j}| < \eta) = 1. \tag{3.8}$$

  Therefore, the p-value converges to 1 as $m \to \infty$. Alternatively if $l \notin D_s$, there exists an item k', such that

  $$\lim_{m \to \infty} p\text{-value} = \lim_{m \to \infty} \mathbb{P}(|\hat{b}_{1,l} - \hat{b}_{k\ ,l}| \geq \eta) = 0. \tag{3.9}$$

  Combining inequalities (3.8) and (3.9) imply that the probability that we misclassify feature l (in terms of whether or not feature l is at the department level) converges to 0 as $m \to \infty$, that is, the DAC consistently identifies whether or not feature l is at the department level.

- Case 3: Feature l is at the cluster level.

  As in the previous cases, we know that as $m \to \infty$, all the coefficients center around their true values with an arbitrarily high probability. Given the number of clusters k, the task is to partition $\{\hat{b}_{i,l} : i = 1, 2, , \cdots, n\}$ into k groups such that the sum of squared Euclidean distances to each group mean is minimized. In general, the high-dimensional k-means algorithm is NP-hard. However, for this specific one-dimensional k-means problem, there exists a dynamic programming algorithm, with a polynomial time complexity, that finds the optimal solution [109]. This implies that the DAC consistently identifies whether or not feature l is at the cluster level, and hence concludes the proof. □

Proof of Proposition 2

(a) The result follows from Theorem 3 in [36].

(b) If $\mathcal{I}_i(\beta_i)$ is diagonal for each i, it means that

$$\frac{\partial^2}{\partial\beta_{i,l}\partial\beta_{i,l}}\left[\sum_{j=1}^{m}\log\mathcal{L}(\beta_i|Y_{i,j},X_{i,j}^1,\ldots,X_{i,j}^d)\right]=0,\qquad\forall l = l'.$$

This implies that for any two different coefficients, the second-order derivative evaluated at the true coefficients is equal to 0. Therefore, for any off-diagonal entries of $\mathcal{I}(\beta)$, we have

$$\begin{aligned}\mathcal{I}(\beta)_{u,v} &= \frac{\partial^2}{\partial\beta_u\partial\beta_v}\left[\sum_{i=1}^{n}\sum_{j=1}^{m}\log\mathcal{L}(\beta|Y_{i,j},X_{i,j}^1,\ldots,X_{i,j}^d)\right]\\ &= \sum_{i=1}^{n}\left[\frac{\partial^2}{\partial\beta_u\partial\beta_v}\sum_{j=1}^{m}\log\mathcal{L}(\beta|Y_{i,j},X_{i,j}^1,\ldots,X_{i,j}^d)\right]=0,\qquad\forall u = v.\end{aligned}$$

To analyze the diagonal terms of the information matrix, we assume that

$$\lim_{m\to+\infty}\frac{1}{m}\frac{\partial^2}{\partial\beta_{i,l}^2}\left[\sum_{j=1}^{m}\log\mathcal{L}(\beta_i|Y_{i,j},X_{i,j}^1,\ldots,X_{i,j}^d)\right]=\frac{1}{\kappa_{i,l}},$$

where $\kappa_{i,l}>0$ is a constant for some item i and feature l. We then have

$$\lim_{m\to+\infty} m\cdot\mathrm{Var}(\hat{b}_{i,l})=\lim_{m\to+\infty}\frac{m}{\frac{\partial^2}{\partial\beta_{i,l}^2}\sum_{j=1}^{m}\log\mathcal{L}(\beta_i|Y_{i,j},X_{i,j}^1,\ldots,X_{i,j}^d)}=\kappa_{i,l}.$$

One can now derive the asymptotic variance of the coefficient under aggregated estimation:

$$\begin{aligned}\lim_{m\to+\infty} m\cdot\mathrm{Var}(\hat{\beta}_{i,l}) &= \lim_{m\to+\infty}\frac{m}{\frac{\partial^2}{\partial\beta_{i,l}^2}\sum_{i=1}^{n}\sum_{j=1}^{m}\log\mathcal{L}(\beta_i|Y_{i,j},X_{i,j}^1,\ldots,X_{i,j}^d)}\\ &= \lim_{m\to+\infty}\frac{m}{n_{i,l}\cdot\frac{\partial^2}{\partial\beta_{i,l}^2}\sum_{j=1}^{m}\log\mathcal{L}(\beta_i|Y_{i,j},X_{i,j}^1,\ldots,X_{i,j}^d)}=\frac{\kappa_{i,l}}{n_{i,l}}.\end{aligned}$$

As a result, we have shown that

$$\lim_{m\to+\infty} m\cdot n_{i,l}\cdot\mathrm{Var}(\hat{\beta}_{i,l})=\kappa_{i,l},$$

where $n_{i,l}$ denotes the number of items that share the same coefficient with item i for feature l. □

Proof of Proposition 7

First, for the aggregated model, we have

$$||X\beta - X\hat{\beta}||_2^2 \quad = \quad ||X\beta - X\beta - X(X'X)^{-1}X'\epsilon||_2^2 = ||P\epsilon||_2^2,$$

where $P = X(X'X)^{-1}X'$ is an idempotent matrix (i.e., a matrix which, when multiplied by itself, yields itself). Since $\epsilon \sim N(0, \sigma^2 I)$, we can write,

$$\frac{||X\beta - X\hat{\beta}||_2^2}{\sigma^2} \quad = \quad \frac{\epsilon'}{\sigma} \quad P \quad \frac{\epsilon}{\sigma} \quad ,$$

where ϵ/σ is a standard normal vector. Based on Section B11.4 in [47], the above quantity follows a χ^2 distribution with degrees of freedom equal to rank(P). By the commutativity property of the trace operator, we have

$$trace(P) = tr(X(X'X)^{-1}X') = tr(X'X(X'X)^{-1}) = d_x,$$

which is the column rank of matrix X. If X has full rank, then d_x represents the number of features under the aggregated model.

Based on Lemma 1 in [68], we can use the tail bound of χ^2 distribution on our mean squared errors. For any $\gamma > 0$, we have the following high probability upper bound:

$$\begin{aligned} &\mathbb{P} \quad \frac{||X\beta - X\hat{\beta}||_2^2}{\sigma^2} - d_x \geq 2 \quad \overline{\gamma d_x} + 2\gamma \quad \leq \exp(-\gamma), \\ \Longrightarrow \quad &\mathbb{P} \quad \frac{||X\beta - X\hat{\beta}||_2^2}{n \times m} \leq \frac{\sigma^2 \left(2\sqrt{\gamma d_x} + 2\gamma + d_x\right)}{n \times m} \quad \geq 1 - \exp(-\gamma). \end{aligned} \tag{3.10}$$

On the other hand, if we use a simple OLS for each item, we obtain n terms of squared errors $||X_i b_i - X_i \hat{b}_i||_2^2$. Each term is a χ^2 distributed variable with degrees of freedom equal to d. Thus, when computing the MSE for the decentralized model, we obtain:

$$\mathbb{P}\left(\frac{{}_{i=1}^{n} ||X_i b_i - X_i \hat{b}_i||_2^2}{n \times m} \leq \frac{\sigma^2 \quad 2 \quad \overline{\gamma\,(nd)} + 2\gamma + nd}{n \times m} \right) \geq 1 - \exp(-\gamma). \tag{3.11}$$

Note that $d_x \in [d, nd]$. Therefore, unless all the features are at the SKU level, we have $d_x < nd$, and thus the bound in (3.10) is tighter than the bound in (3.11). Consequently, we can achieve a smaller MSE for the aggregated model relative to the decentralized model under the same level of confidence, $\exp(-\gamma)$.

In addition, we can provide a high probability lower bound for the decentralized model. As shown in [68], we have

$$\mathbb{P}\left(\frac{{}_{i=1}^{n} ||X_i b_i - X_i \hat{b}_i||_2^2}{n \times m} \geq \frac{\sigma^2 \quad nd - 2 \quad \overline{\gamma\,(nd)}}{n \times m} \right) \geq 1 - \exp(-\gamma). \tag{3.12}$$

If we compare the upper bound in (3.10) to the lower bound in (3.12), we can solve for the sufficient condition under which the aggregated model outperforms the decentralized model with high probability:

$$\begin{aligned} & \frac{\sigma^2\left(2\sqrt{\gamma d_x}+2\gamma+d_x\right)}{n\times m} \leq \frac{\sigma^2 \quad nd-2 \quad \overline{\gamma\,(nd)}}{n\times m}, \\ \implies \quad & d_x^2-(2A+4\gamma)d_x+A^2\geq 0, \end{aligned}$$

where $A = nd - 2\sqrt{\gamma nd} - 2\gamma$. As a result, the above inequality holds when

$$d_x \leq nd - 2 \qquad \overline{\gamma nd} + \quad \overline{\gamma nd - 2\gamma\sqrt{nd} - \gamma^2}$$

and this concludes the proof of Proposition 7. □

Chapter 4

Peer-to-Peer Lending

4.1 Introduction

In recent years, online platforms and peer-to-peer services have burgeoned with applications in most industries. Examples include transportation, delivery-services, private tutoring, and home-maintenance tasks, just to name a few. The financial sector has also witnessed a similar disruption with the apparition of crowdfunding and peer-to-peer lending platforms. At a high level, crowdfunding competes with traditional venture capital funds, and peer-to-peer lending aims to challenge the personal-loan industry. As the recent Bloomberg article in [69] puts it: “Personal loans surged to a record this year [2018] and are the fastest-growing U.S. consumer-lending category, according to data from credit bureau TransUnion. Outstanding balances rose about 18% in the first quarter to $120 billion. Fintech companies originated 36% of total personal loans in 2017 compared with less than 1% in 2010, Chicago-based TransUnion said.”

Peer-to-peer lending is the practice of lending money to individuals via online services that match anonymous lenders with borrowers. Since these companies typically operate online, they can often provide service at a lower cost than traditional financial institutions. Consequently, borrowers can borrow capital at lower interest rates, and lenders can potentially earn higher returns relative to the investment products offered by banking institutions. However, there is of course the risk that the borrower defaults on his or her loan. Interest rates are usually set by an intermediary platform on the basis of analyzing the borrower’s credit-worthiness (based on features such as credit score, employment status, annual income, debt-to-income ratio, and credit history). The platform often categorizes loans using a grading system (e.g., A to F, where A loans are considered the safest and hence have low interest rate). The intermediary platform generates revenue by collecting a one-time fee on funded loans (from borrowers) and by charging a loan servicing fee to investors.

The peer-to-peer lending industry has now become a global phenomenon. In the U.K., it started in February 2005, and as of 2015, U.K. peer-to-peer lenders collectively lent over £3 billion. In the U.S., it started in February 2006. In 2008, the Securities and Exchange Commission (SEC) required peer-to-peer companies to register their offerings as securities, pursuant to the Securities Act of 1933. As of July 2018, there are at least six reputable peer-to-peer lending companies in the U.S.[1] Some of these platforms are reporting great financial performance, whereas others are not, or even went bankrupt. This is true in most countries. For example, in China, in 2016 there were more than 4,000 platforms, but half of them had already suspended operations by 2018.

When a firm becomes successful, an important milestone is to decide whether or not to go public. Interestingly, according to research from The Atlantic, there are half as many public companies listed on U.S. stock exchanges in 2018 as there were in 1997.[2] Obviously, the decision of whether to file an initial public offering (IPO) is intricate and involves several factors (this topic is beyond the scope of this chapter). In the context of peer-to-peer lending, several platforms have decided to go public in the last few years. Examples include Funding Circle in the U.K., LendingClub in the U.S., and PPDAI Group in China. At the same time, several other platforms still remain privately held to this day (e.g., Zopa in the U.K., Prosper Marketplace in the U.S., and WeLab in China). Assuming that a peer-to-peer lending platform has decided to go public, we are interested in studying the impact of its IPO. Manipulative activities in anticipation of an IPO, though prohibited by SEC regulations, are not unusual in practice. According to [3], among 908 IPOs during 1998–2000, 173 were subjects to class action lawsuits. Also, in recent years, extensive media reports have covered potential pre-IPO manipulations (e.g., Bitmain).[3] With this in mind, our goal is to investigate whether such activities are present in the peer-to-peer lending industry. Since the business model of such platforms relies heavily on commissions, the number of issued loans directly translates into revenue. Thus, in anticipation of its IPO, a platform may have an incentive to increase its number of issued loans to inflate its predicted market capital value. This strategy can pay off in the short term but may be detrimental in the long run by adversely affecting the platform's reputation. Indeed, if the platform starts accepting low-quality loans, investors can lose trust and decide to stop investing in the platform.

In this chapter, we use a comprehensive dataset from two large peer-to-peer lending platforms (denoted P_1 and P_2).[4] P_1 and P_2 operate in the same country and are the two dominant platforms in terms of market share and volume. In 2017, these two platforms jointly issued the equivalent amount of 9.46 billion U.S. dollars. Moreover, these two platforms have a similar operating procedure for both borrowers and lenders. We are facing an interesting situation where P_1 went public by filing an IPO, while P_2 still remained privately held. We exploit this empirical environment to carefully infer the causal effect of IPOs on peer-to-peer lending platforms. It is worth highlighting that our analysis only became possible recently as one needs to wait until all loans issued around the IPO date reach their maturity.

The main motivation behind this chapter is to answer the following questions:

- When filing its IPO, did P_1 alter its decisions by accepting different types of loans or by softening its

[1] https://www.investopedia.com/articles/investing/092315/7-best-peertopeer-lending-websites.asp

[2] https://www.theatlantic.com/magazine/archive/2018/11/private-inequity/570808

[3] https://cryptoslate.com/bitmain-ipo-is-the-unicorn-mining-giant-misleading-investors/

[4] We mask the platforms' names for neutrality. We also mask the exact time events and the currency involved in the loans.

borrowers' requirements?

- What was the impact of P_1's IPO on investors and on borrowers?

- Which time events (related to the IPO) were the most impactful (e.g., IPO filing and actual IPO date)?

Empirical methods: To answer the above questions, we estimate several econometric models. We present rigorous empirical analyses at four time events: first IPO rumor, filing date, actual IPO, and quarterly report release. For each event, we test the robustness of our findings by considering a multitude of approaches and model specifications. Given that our data include both accepted and rejected loans, we propose to estimate a two-stage regression model to correct for the sample selection bias (see more details in Section 4.3.1). We perform both self-platform and cross-platform analyses. Our goal is to use the data from P_2 as a control group to measure the impact of P_1's IPO. Given that P_1 and P_2 are not perfectly comparable, we use propensity score matching (PSM) to select a subset of loans that share similar features across both platforms. We then estimate a difference-in-differences model using the PSM samples. Finally, we implement the synthetic control method as an additional approach to validate our results.

Main findings: We find that several performance metrics were indeed affected by the IPO, especially around the IPO filing date. Specifically, when comparing P_1 loans before and after the IPO filing (while using P_2 loans as a control), we observe the following:

- The loans' performance (default rate and return) significantly decreased. Based on our synthetic control results, the average default rate increased from 10.9% to 13.96%, and the average (annual) return decreased from 3.18% to 2.65%.

- Borrowers' requirements (credit score and annual income) also diminished. Our synthetic control results suggest that the average credit score decreased from 702 to 696.3, and the average annual income decreased from 73,750 to 70,083.

- The acceptance rate of P_1 inflated from 12.70% to 13.18% (the difference is statistically significant at the 0.01 level).

In addition, the largest performance decrease was observed for riskier loans (i.e., loans with a high interest rate and a low grade), whereas the credit score decreased the most for safer loans. These findings suggest that P_1 may have altered some of its decisions in anticipation of its IPO by lowering its requirements for accepting loans. Namely, it is plausible that P_1 issued loans with lower credit scores and annual incomes right after its IPO filing. This strategy can potentially allow P_1 to increase its short-term revenue in view of boosting its market capital value. Interestingly, after the actual IPO date, all performance metrics returned to their initial levels.

4.1.1 Related Literature

We next review two streams of literature related to this chapter: peer-to-peer lending and IPO (in finance). We then briefly discuss the econometric methods used in our analysis.

Peer-to-peer lending: A large number of papers in the context of peer-to-peer loans focus on implementing machine learning algorithms to predict credit risk (or default probability) ([66, 79, 24]). These papers use datasets from a single platform (they often use the publicly-available data from LendingClub[5]), and apply learning methods to predict borrowers' default probability. Other papers examine the funding dynamics of peer-to-peer lending platforms (see, e.g., [118, 111, 72]). For example, [111] develop a game-theoretic model to compare a multi-unit uniform-price auction versus a posted-price mechanism in the context of peer-to-peer lending. The authors empirically test their model and find that under platform-mandated posted prices (for interest rates), loans are funded with higher probability. Furthermore, the interest rates are also higher under a posted-price mechanism. [72] consider a setting where borrowers' online friendships (e.g., referrals) can act as a signal of credit quality, which may increase the probability of funding and decrease the interest rate. Finally, several papers study how non-standard information can impact the platform. For example, [58] examine how to leverage non-expert reviews to evaluate borrowers' quality (i.e., likelihood of defaulting), whereas [96] investigates how usury laws (i.e., market regulations that cap interest rates) affect loan performance and funding probability. This chapter differs from previous work in several aspects. First, while the aforementioned studies use data from a single platform, we combine the datasets from two large platforms and conduct cross-platform analyses. Ultimately, our goal is to use one platform as the treatment group and the other as a control. Second, most of the previous papers have focused on the interaction between the platform and the borrowers. In our case, we consider several external events (IPO-related activities) and investigate the impact of these events on both sides of the market (investors and borrowers).

Another stream of literature focuses on loan portfolio optimization. For example, [101] consider the problem of optimally selecting a large portfolio of risky loans. The optimal selection requires solving a high-dimensional nonlinear integer program, and the authors develop an approximate optimization approach that yields an asymptotically optimal portfolio. [83] proposes a methodology to analyze the risk and return of large loan portfolios. [100] develop and test efficient numerical methods for the analysis of large pools of loans as well as asset-backed securities. The authors prove a law of large numbers and a central limit theorem for the pool-level risk.

IPO: Initial public offerings have been studied in the finance community for decades. Several papers focus on empirical analyses of pre-IPO pricing strategies and post-IPO outcomes (e.g., stock performance relative to the market). For example, in their seminal paper, [5] develop a model based on empirical evidence and suggest that under certain circumstances, firms with a favorable prospect find it optimal to underprice their initial price per share. [19] study how the underwriter's reputation (i.e., group of investment banks who exercise the IPO) affects the post-IPO stock performance. More relevant to this chapter, [3] find that

[5] `https://www.lendingclub.com`

some underwriters created artificial excess demand in anticipation of the IPO, leading to manipulated price levels in the immediate aftermarket. [59] investigate the change in firms' operating performance as they transit from private to public ownership. A number of studies investigate earnings management during the IPO year and typically find that firms manage earnings upward before the IPO to boost post-IPO stock returns (see, e.g., [43, 104, 31]). Several recent papers study the impact of government interventions on firms' behavior. For example, [27] find that suspension-induced delays of IPOs substantially reduce innovation activities, measured by the number of patents (and their quality). Other studies consider how firm's decisions impact the company performance. [87] develop a stochastic model to decide the optimal IPO timing and show that the firm profitability should decline after the IPO. [9] investigates the effects of going public on firm's innovation and show that the quality of internal innovation declines after the IPO. To our knowledge, however, the impact of IPOs on peer-to-peer lending has not been studied before.

Methods: In this chapter, we estimate several commonly used empirical tools. Apart from t-tests, ordinary least squares, propensity score matching, and difference-in-differences, we also use a Heckman correction and synthetic control. The sample selection bias correction was first introduced by [50] and has been used in many empirical studies. This method has also been combined with other techniques. For example, in [45] and [11], the authors implement a two-stage Heckman correction combined with a difference-in-differences regression, as in our setting. Several operations management researchers have recently used the Heckman method (see, e.g., [51]). Synthetic control has been extensively used in economics and finance since the seminal paper in [1], which estimated the effect of California's tobacco control program using 38 other U.S. states to construct a control group. This method uses several observed groups to construct a control that resembles the treatment group as much as possible. In our setting, given that we only have access to a single control group, we propose to split the data from this group into several subgroups and construct the synthetic control based on these subgroups.

Structure of the chapter. The rest of the chapter is organized as follows. In Section 4.2, we discuss our data and the metrics we consider. The econometric methods used in our analysis are discussed in Section 4.3. Our results and insights are presented in Sections 4.4 and 4.5, respectively. In Section 4.6, we conduct several tests to showcase the robustness of our results. Finally, our conclusions are reported in Section 4.7. Additional tables and figures are relegated to the Appendix.

4.2 Data and Metrics

4.2.1 Data

We have access to the data generated by two large peer-to-peer online platforms, denoted P_1 and P_2. For both P_1 and P_2, we have detailed information on all issued loans starting from the platform launch until the second quarter of 2018. These data files include various features on each loan, such as (fixed) interest rate, borrower's credit score, debt-to-income ratio, and annual income. For P_1, in addition to issued loans, we also have the data on the loan applications that did not meet the platform's credit underwriting policy. We refer to such listings as rejected loans, and to the combination of accepted and rejected loans as our listing data. On the other hand, loan applications that did not pass P_2's initial pre-screening are not included in our data (whereas for P_1, all applications are observed). This difference will affect our analysis on the acceptance rate, as we discuss later.

Loans from both platforms have only two possible length terms: 36 months and 60 months. Since the events related to P_1's IPO occurred recently, 60-month loans issued during that time have not reached maturity yet. Thus, to avoid the potential survivorship bias (see, e.g.,[15]), we focus on 36-month loans (which represent 68.3% of the loans issued over the period of interest). In particular, our data span a period of two years (the same 24 months for both platforms). Specifically, we have 416,667 loans for P_1 and 179,463 loans for P_2. Each observation is a loan and includes the following features: time of application, loan amount, length term (all are 36 months), offered interest rate, grade assigned by the platform, total payment received, and loan status (e.g., fully paid, defaulted). Regarding the borrower, we observe attributes such as employment length, self-reported annual income, credit history length, credit score, DTI (debt-to-income) ratio, and number of derogatory records. We report the basic statistics of the most important variables in Table 4.1 (for both P_1 and P_2). As one can see, both platforms are somewhat comparable in most attributes, but we will further subsample loans with similar features in Section 4.4.2.

In our data, except for annual income, all variables have a reasonable range so that the data do not need to be filtered out. The annual income, however, could go as high as 55 standard deviations from the mean. To ensure data quality, we remove the top and bottom two percentiles of observations with respect to annual income for each time period.

Table 4.1: Summary statistics for P_1 and P_2

Variable	P_1		P_2	
	Mean	Standard deviation	Mean	Standard deviation
Loan amount	12,817	7,840	12,327	7,462
Credit score	697.15	30.35	706.85	37.02
Annual income	71,537	35,315	70,727	34,594
Credit history	194.17 months	87.89 months	217.49 months	99.32 months
DTI	18.1%	8.3%	27.0%	11.75%

Since our focus is on P_1's IPO-related events, we consider loans that were issued 36 months before the IPO (so that they are terminated by now and we have full information). It is worth mentioning that this analysis has become possible only recently as we have passed the natural maturity of 36-month loans issued after the IPO. We focus on the following four time events:

1. **$\mathbf{T_1}$ - First IPO rumor.** This is the first time that there was a credible rumor in the news that P_1 will go public.

2. **T_2 - Official filing.** This is the date at which P_1 filed its IPO on the stock exchange.

3. **T_3 - Actual IPO.** This corresponds to the time when the IPO has actually materialized (it takes several weeks between filing and materializing).

4. **T_4 - Second quarterly report release.** This is the date when P_1 released its second quarterly report—an obligation for all public companies (note that the first quarterly report was released very close to the actual IPO and is thus omitted).

The exact times of these four events were obtained using the online platform Factiva.[6] Factiva is a news platform that can be used to track the exact dates of historical events. We treat each major historical event as a natural intervention (or treatment). Conveniently, all four events are equally-spaced by a period of three to four months.

Our goal is to examine whether these events had an impact on the outcome variables of interest, which are discussed next.

4.2.2 Metrics

We consider three different categories of outcome variables.

Performance metrics: The first category reflects the performance of the platform from investors' perspective. We focus on two variables: default rate and return. Since we only consider loans that are terminated at the time of analysis, each loan ends in one of two statuses: fully paid or defaulted/charged-off. A loan is labeled as charged-off if the payment is delayed by more than 150 days or if the platform decided that the loan will not be paid back. We then define the default rate as the ratio of defaulted loans to all issued loans during a specific period.

The return definition is more intricate than it appears. Peer-to-peer investments are in nature a type of fixed-income investment that generates a monthly fixed amount of cash regardless of inflation or the market risk-free rate. However, unlike traditional bonds, a loan can default or can be early-repaid at any time during its term, making the return calculation more delicate. We use the following (annual) return definition, labeled as Method 1 (M1):

$$\frac{p-f}{f} \times \frac{12}{t},$$

where f is the total loan funded amount, p is the total amount recovered from the loan, and t is the loan term length in months (in our case 36). M1 assumes that once the loan is paid back, the investor is not considering any re-investment. We present our analysis using M1 and showcase the robustness of our results under two alternative return definitions (see Section 4.6.3).

Borrowers' attributes: The second category is associated with borrowers' attributes. Specifically, we study the following set of features:

[6] `https://www.dowjones.com/products/factiva/`

- Upper value of the credit score,[7]
- Annual income, and
- Length of credit history (in months).

It is important to note that the values of all features were recorded at the time of loan application. The changes in these values over time reflect the internal decision criteria of the platform. For example, if we observe that the average credit score of applicants decreases over time, it may suggest that the platform's acceptance criterion was softened.

Platform decisions: The third category reflects the platforms' operational decisions. In particular, we focus on the acceptance status of loans' applications. We cannot compute this metric for P_2 as we do not have access to this information. For P_1, we compute the average acceptance rate and study its variation over time. The average acceptance rate is defined as the ratio of the number of accepted loans to the total number of applications during a specific period.

In summary, we study six dependent variables (DVs) which are summarized in Table 4.2.

Table 4.2: The dependent variables used in this chapter

Dependent variable	Range of values
Default rate	$[0, 1]$
Return	$[-1, 0.3]$
Credit score	$[600, 850]$
Annual income	$[22,000, 300,000]$
Credit history (months)	$[6, 780]$
Acceptance status	$[0, 1]$

4.3 Econometric Methods

In this section, we discuss the statistical and econometric techniques used in this chapter. Let Y_i denote the value of a particular dependent variable for loan i (e.g., return) and X_i denote the other explanatory variables related to loan i (e.g., loan amount).

4.3.1 Self-Platform Analysis

T-Test

This is one of the most common techniques to compare two samples of data. It can help determine whether the average difference between two groups is statistically significant (i.e., different from zero). Denote the

[7] The credit score is reported as a range with lower and upper values. Since the lower value appears less trustworthy, we use the upper value in our analysis.

average values of the two groups (in our case P_1 and P_2) as $\bar{Y}_1$ and $\bar{Y}_2$. Similarly, denote the standard deviations as s_1 and s_2 and the number of observations as n_1 and n_2. We can then compute the following t-statistic:

$$t = \frac{\bar{Y}_1 - \bar{Y}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}. \tag{4.1}$$

Next, one can compare the t-statistic value from Equation (4.1) to a standard table to determine whether the statistical significance threshold is attained. If the p-value is smaller than the threshold (e.g., 0.01, 0.05, 0.1), we reject the null hypothesis and conclude that two groups have different average values.[8]

Two-Stage Regression Analysis

Ordinary least squares (OLS) is a fundamental method to analyze correlation and causal effects. In this chapter, we estimate OLS models for each platform separately. Estimating OLS specifications allows us to test for causality between the dependent variable and the treatment event while controlling for external factors. For binary dependent variables (e.g., default rate), we use the linear probability model, which assumes that the outcome probability depends linearly on the explanatory variables. We choose OLS rather than a logistic regression due to the following two reasons. First, OLS leads to interpretable coefficients that can be used to quantify the impact of interest. Second, in the cross-platform analysis (see Section 4.3.2), we need to use clustered standard errors given that the loans can be split into two clusters (corresponding to P_1 and P_2). In this case, model errors of loans from the same cluster may be correlated. From an implementation standpoint, it is easier to cluster standard errors using a linear model. Nevertheless, most of our qualitative results still hold when using a logistic regression (with regular standard errors).

Sample selection bias: A major issue in our dataset is a sample selection bias. Specifically, issued loans were not randomly drawn from the entire population of loan applications. Instead, issued loans were screened and selected based on internal requirements set by the platform. Therefore, if we want to study the effect of an event on a dependent variable, we need to carefully correct the model for this selection bias. As is common in many empirical settings, we overcome the issue by using the Heckman correction [50], which leads to a two-stage model. Suppose we use the following OLS equation to study the effect of explanatory variables X_i on the dependent variable Y_i for a given platform:

$$Y_i = X_i\beta + \epsilon_i.$$

As mentioned, the inclusion of observation i is not random. We assume the probability that listing j is issued (or accepted) by the platform follows a probit model with the following specification:

$$\Pr(A_j = 1 \mid Z_j) = \Phi(Z_j\gamma), \tag{4.2}$$

[8] As is common, we use *** for a significance level below 0.01, ** for the level between 0.01 and 0.05, and * for the level between 0.05 and 0.1.

where A_j indicates the status of listing j (equals 1 if the listing becomes issued and 0 otherwise), Z_j is the set of explanatory variables, and γ is the vector of coefficients to be estimated. Equation (4.2) is called the first stage of our regression model. In our setting, Z_i includes several explanatory variables that are shared across issued and non-issued observations. For an issued observation (say k), the vector X_k will not overlap with the attributes in Z_k. The covariates used in our analysis are reported in Section 4.4.1. After estimating γ, we can write the second-stage equation for any issued loan i as

$$Y_i = X_i\beta + \rho\lambda(Z_i\gamma) + \epsilon_i, \tag{4.3}$$

where $\lambda(\cdot)$ is the inverse Mills ratio evaluated at $Z_i\gamma$, and ρ is a coefficient to be estimated. The inverse Mills ratio is the ratio of the probability density function to the cumulative distribution function. Equation (4.3) is derived under the independence and normality assumptions of the error terms (for more details, see [50]).

4.3.2 Cross-Platform Analysis

Our econometric tools were designed to estimate outcome variables for each platform separately. For example, we can infer that a specific treatment (e.g., P_1's IPO filing) had an effect on P_1 but not on P_2. However, one could argue that a fairer comparison would be based on cross-platform analysis by using P_1 as treatment and P_2 as control. In this spirit, we consider two approaches: difference-in-differences (with and without propensity score matching) and synthetic control.

Difference-in-Differences

As discussed, to rigorously study the treatment effect of the events, we want to combine both datasets and use P_2 as a control group. To this end, we implement the difference-in-differences (diff-in-diffs)—a method designed to study the differential effect of a treatment. Accordingly, it calculates the effect of an event on a dependent variable by comparing the average change in the outcome of the treated group relative to the control group. Since the combined dataset still suffers from sample selection bias, we implement the Heckman correction for the cross-platform analysis (a similar model was implemented in [45, 11]). In this case, the first stage probit model is given by:

$$\Pr(A_j = 1 \mid Z_j, P_j) = \Phi(Z_j\gamma + P_j\delta),$$

where P_j represents listing j's platform, and Z_j contains the set of common explanatory variables present in the listings of both platforms. For the second stage, we consider the following specification:

$$Y_i = X_i\beta_0 + \rho\lambda(Z_i\gamma + P_i\delta) + \beta_1 \cdot \text{time}_i + \beta_2 \cdot \text{intervention}_i + \beta_3 \cdot \text{time\_intervention}_i + \epsilon_i, \tag{4.4}$$

where X_i denotes the set of explanatory variables (see Section 4.4). The variables time_i and intervention_i are defined as follows:

$$\text{time}_i = \begin{cases} 0 & \text{if loan } i \text{ was issued before the treatment event,} \\ 1 & \text{otherwise.} \end{cases}$$

$$\text{intervention}_i = \begin{cases} 0 & \text{if loan } i \text{ belongs to the control group } (P_2), \\ 1 & \text{if loan } i \text{ belongs to the treated group } (P_1). \end{cases}$$

The variable $\text{time\_intervention}_i$ corresponds to the product of time_i and intervention_i. We are interested in the coefficient β_3, which captures the impact of the treatment effect on P_1 relative to P_2. The specific variables and other implementation details can be found in Section 4.4.2. Also, in Section 4.4, we will report the estimated values of β_3 (along with its significance level) for each dependent variable.

Our proposed specification aims to mitigate the effects of extraneous factors and selection bias while allowing us to carefully examine the impact of the treatment (i.e., P_1's IPO-related events). While the diff-in-diffs method allows for both populations to have different outcome levels (by focusing on the relative post-intervention variation), one may still argue that P_2 is not an appropriate control group. Indeed, both platforms may attract different types of borrowers and investors. In addition, the payment structure may differ (e.g., different commission levels). In other words, is it legitimate to use the loans from P_2 as a control group for our analysis? To mitigate this issue, we carefully select a subset of P_1 loans that share similar features with P_2 loans (given that the data size of P_2 is smaller, we select a subset of P_1 loans while using the entire data from P_2). To do so, we implement a propensity score matching (PSM) estimator as follows:

1. We run a logistic regression on the binary platform variable ($PL_i = 1$ if loan i belongs to P_1 and 0 otherwise) using all common features across both platforms as explanatory variables.

2. We then obtain the predicted propensity score for each loan, which represents the probability that loan i belongs to P_1.

3. We match each loan from the control group (P_2) to a member of the treatment group (since we have a larger number of treated loans) using a matching algorithm (e.g., nearest-neighbor).

4. We verify that the covariates are balanced across the treatment and control groups.

The PSM method would then be combined with our diff-in-diffs specification, as done in several previous studies (see, e.g., [102, 114]). After Step 4 above, we obtain a new dataset, which will be used as our treatment group. In Section 4.4, we will implement the diff-in-diffs method both with and without PSM.

Synthetic Control

Synthetic control has been extensively used in economics and finance since the seminal paper in [1]. It is built upon the diff-in-diffs model with the following two changes: (1) it allows for time-varying individual heterogeneity, and (2) it takes a data-driven approach to select the control group. It is also more appealing from a visualization perspective as it allows us to observe the trend of the treatment effect before and after the intervention.

We next discuss in greater detail the synthetic control approach.[9] Suppose we observe $J+1$ groups of data over $t = 1, \ldots, T$ periods, where Group 1 is treated and Groups $\{2, \ldots, J+1\}$ are non-treated. An intervention (i.e., treatment event) occurs at period $T_0 + 1$, where $1 < T_0 + 1 < T$ and only affects Group 1. Let $Y_{i,t}^N$ denote the outcome value for Group i at time t in the absence of intervention, and $Y_{i,t}^I$ denote the outcome value for Group i that would have been observed at time t if Group i had been treated. The goal of the synthetic control method is to estimate the treatment effect over time for the treated unit (i.e., Group 1), that is,

$$\alpha_1 = (\alpha_{1,T_0+1}, \ldots, \alpha_{1,T}).$$

In particular, for $t > T_0$, we have:

$$\alpha_{1,t} = Y_{1,t}^I - Y_{1,t}^N = Y_{1,t} - Y_{1,t}^N.$$

Note that we only observe $Y_{1,t} = Y_{1,t}^I$. Therefore, we need to construct the unobserved counterfactual $Y_{1,t}^N$. This term is assumed to follow the factor model for each Group i:

$$Y_{i,t}^N = \delta_t + \theta_t Z_i + \lambda_t \mu_i + \epsilon_{i,t},$$

where $\epsilon_{i,t}$ is a zero-mean shock and

1. δ_t is an unobserved common factor;

2. Z_i is a vector of observed covariates unaffected by the intervention, and θ_t is a vector of parameters to be estimated; and

3. μ_i is a vector of unobserved common factors, and λ_t is a vector of parameters to be estimated.

The basic idea behind synthetic control relies on assigning weights to the different control groups so that a "synthetic" unit matches Z_i and some pre-treatment $Y_{1,t}$. As a result, μ_i will be automatically matched. Let $W = (w_2, \ldots, w_{J+1})$ be a vector of non-negative weights that sums up to 1. The outcome of the synthetic control method at time t can then be written as:

$$Y_{W,t} \quad = \quad \sum_{j=2}^{J+1} w_j Y_{j,t} = \delta_t + \theta_t \left(\sum_{j=2}^{J+1} w_j Z_j \right) + \lambda_t \left(\sum_{j=2}^{J+1} w_j \mu_j \right) + \left(\sum_{j=2}^{J+1} w_j \epsilon_{j,t} \right).$$

The major problem in our case is that we only have a single control group (platform P_2), whereas the synthetic control method requires multiple control groups. To solve this issue, we split the data from P_2 into several subgroups based on different ways of splitting. Then, we use each subgroup as an individual control group. Note that splitting randomly would not help creating variation across the subgroups. The results of this method are presented in Section 4.4.2.

[9] To estimate our model in Section 4.4, we use the gsynth R package.

4.3.3 Moderating Factor

To refine our understanding on the relationship between the dependent variable and the treatment event, we consider the risk level of loans as a moderating factor. This allows us to check whether the effect is universal or if it depends on the loan type.

We split the different loans according to their risk level. We capture the risk level of a loan using one of the following two criteria: the interest rate or the grade assigned by the platform. Both criteria reflect the risk level of loans. The lower the grade (or the higher the interest rate), the riskier the loan. For single-platform analysis, we define the risk level of loan i as follows:

$$\text{Risk Level}_i = \begin{cases} \text{Safe} & \text{if Int Rate}_i < P_{33}, \\ \text{Medium} & \text{if } P_{33} \leq \text{Int Rate}_i < P_{66}, \\ \text{Risky} & \text{if Int Rate}_i \geq P_{66}. \end{cases}$$

Here, P_{33} and P_{66} are the 33-rd and 66-th percentiles of the interest rate distribution across all loans issued by the platform. For cross-platform analysis, however, following the same procedure can be problematic as the percentiles of the two platforms are different. To address this issue, we define the risk levels using pre-defined threshold values (e.g., 10% and 15%) to ensure fair comparisons:

$$\text{Risk Level}_i = \begin{cases} \text{Safe} & \text{if Int Rate}_i < 0.1, \\ \text{Medium} & \text{if } 0.1 \leq \text{Int Rate}_i < 0.15, \\ \text{Risky} & \text{if Int Rate}_i \geq 0.15. \end{cases}$$

For robustness purposes, we will test several threshold cutoffs (see Section 4.6.2).

4.4 Results

For conciseness, we report all the detailed results for T_2, that is, the IPO filing event. The results for other time events (T_1, T_3, and T_4) are summarized in Section 4.4.3. Consider the treatment at time T_2. Recall that this treatment was applied to P_1 but not to P_2. Our goal is to examine the treatment effect on the six DVs (dependent variables) from Table 4.2. To do so, we consider each DV both before and after the treatment. In this section, we use a time window of three months before and three months after (we test the robustness of our results by considering a larger time window in Section 4.6.1).

4.4.1 Self-Platform Results

T-Test

We perform a t-test based on Equation (4.1) using the set of loans issued by each platform before and after T_2. The results for all six DVs are summarized in Tables 4.3.[10]

[10] As mentioned, we do not have the acceptance rate data of P_2.

Table 4.3: T-test results for P_1 and P_2

Variable	P_1			P_2		
	Value before	Value after	P-value (significance)	Value before	Value after	P-value (significance)
Acceptance	12.70%	13.18%	7.33e-09***	N/A	N/A	N/A
Default	13.24%	14.06%	4.71e-04***	15.12%	15.00%	0.7322
Return	3.06%	2.63%	4.54e-19***	2.97%	2.87%	0.2382
Credit score	697.72	695.14	2.21e-36***	707.20	707.87	0.0658*
Annual income	70535.81	69281.28	7.43e-08***	71942.54	72341.62	0.2345
Credit history	190.81	193.10	1.04e-04***	222.24	223.26	0.2765

As we can see from Table 4.3, the acceptance rate of P_1 increased from 12.70% to 13.18% after the intervention (i.e., after the IPO filing). Regarding the performance of P_1, the average default rate increased and the return dropped. This suggests that the performance of P_1 loans may have suffered from the IPO filing. In contrast, P_2 loans did not have any significant change in default rate and return during this period. These preliminary results suggest that P_1's IPO filing had an impact on P_1 loans. The rest of this chapter aims to investigate this effect more carefully.

To examine whether P_1 has accepted "worse" loans after its IPO filing, we analyze three borrowers' features: credit score, annual income, and credit history length. As we can see, while P_2 loans did not have a significant change in any of these variables (the credit score variation is significant at the 0.1 level, but the magnitude of the difference is small), the average credit score and annual income of P_1 loans decreased. Note that the credit length history increased. However, when compared to the two other attributes, credit history is, to a lesser extent, related to the quality of the borrower. Specifically, it is not clear whether a borrower with a higher credit history is "safer" or "riskier." On the other hand, the credit score and annual income are two commonly used features to determine the borrower's quality. We observe that these two features dropped for P_1. This implies that P_1 softened its screening criteria by accepting more loans and reducing borrowers' requirements (credit score and annual income).

To refine the results of the general t-test, we consider using the moderating factor discussed in Section 4.3.3. Using the same procedure, we run our test on each subgroup separately (the subgroups were obtained based on interest rate percentiles). The results for Risky/Average/Safe loans for P_1 and P_2 are reported in Tables 4.4-4.6.

Table 4.4: T-test results for P_1 and P_2 (Risky loans)

Variable	P_1			P_2		
	Value before	Value after	P-value (significance)	Value before	Value after	P-value (significance)
Default	20.46%	23.16%	3.99e-08***	23.46%	23.46%	0.9987
Return	3.36%	2.39%	1.31e-18***	3.61%	3.36%	0.2152
Credit score	682.24	683.69	4.65e-11***	684.29	684.32	0.9345
Annual income	63756.46	61057.07	7.78e-13***	66139.68	66230.49	0.8725
Credit history	184.25	174.62	8.98e-22***	210.89	212.88	0.2172

The results for Risky loans are similar to the general results: no significant change is found for P_2, whereas for P_1, the loan performance and borrowers' attributes dropped. In this case, both annual income and credit history decreased. Although the average credit score had a significant increase based on the t-test, the magnitude remains small.

Table 4.5: T-test results for P_1 and P_2 (Average loans)

Variable	P_1			P_2		
	Value before	Value after	P-value (significance)	Value before	Value after	P-value (significance)
Default	12.77%	13.50%	0.0742*	14.64%	14.08%	0.3421
Return	3.20%	3.05%	0.0497**	2.91%	2.97%	0.6903
Credit score	691.00	689.00	2.12e-13***	701.29	701.74	0.3404
Annual income	70535.81	69281.28	6.21e-11***	71359.41	72241.75	0.1373
Credit history	191.36	188.61	0.0069***	225.44	223.02	0.1455

For Average loans, there is also a strong signal that filing the IPO had a negative impact on both the performance and quality of P_1 loans. The default rate increased (significant at the 0.1 level), and the return decreased. All three borrowers' attributes (credit score, annual income, credit history) decreased significantly. Again for P_2, all DVs had statistically insignificant changes.

Table 4.6: T-test results for P_1 and P_2 (Safe loans)

Variable	P_1			P_2		
	Value before	Value after	P-value (significance)	Value before	Value after	P-value (significance)
Default	6.34%	6.54%	0.4937	7.94%	8.08%	0.7543
Return	3.20%	3.05%	0.0212**	2.45%	2.35%	0.2373
Credit score	720.30	710.47	2.18e-129***	733.56	735.11	0.0149**
Annual income	78985.71	78943.94	0.9196	77775.16	78039.65	0.6409
Credit history	196.96	213.20	7.14e-59***	229.62	233.01	0.0332**

For Safe loans, the most striking result is that the average credit score of P_1 decreased by close to 10 points after the IPO filing. This is surprising as the average credit score remains stable throughout our entire dataset. The drop in credit score is compensated by an increase in credit history. However, as mentioned before, credit history is not as clear of an indication as the credit score. Once again, the return significantly dropped for P_1 but not for P_2.

Regression Results

To complement the single-platform analysis, we next estimate regression models. As before, for the acceptance status variable, we only have data for P_1. The regression model for acceptance status is given by:

$$\text{Accept\_Status}_i = \beta_0 + \beta_1 \cdot \text{loan\_amnt}_i + \beta_2 \cdot \text{dti}_i + \beta_3 \cdot \text{after\_ref}_i + \epsilon_i, \tag{4.5}$$

where loan_amnt_i is the total amount of loan i. In this case, given that we use the data from the entire population of applicants, we do not need to apply the Heckman correction, and thus there is no inverse Mills ratio in Equation (4.5). We estimate the model using OLS and obtain $\beta_3 = 0.0039$ with a standard error of 0.001. The p-value suggests that the coefficient is positive and statistically significant. This result confirms that after the IPO filing, P_1's acceptance rate increased.

For other dependent variables, we need to implement the two-stage regression model to correct for the sample selection bias. The first stage equation is a probit model:

$$\Pr(A_j = 1) = \Phi(\gamma_0 + \gamma_1 \cdot \text{loan\_amnt}_j + \gamma_2 \cdot \text{dti}_j + \gamma_3 \cdot \text{emp\_length}_j), \tag{4.6}$$

where emp_length_j represents the number of years that listing j's applicant was employed (at the time of application). We use the three above variables (loan_amnt_j, dti_j, emp_length_j) in the first-stage equation because these are the only features present in both issued and non-issued listings. For the second stage, we estimate the following regression equation:

$$Y_i = \beta_0 + \beta_1 \cdot \text{pub\_rec}_i + \beta_2 \cdot \text{mills}_i + \beta_3 \cdot \text{after\_ref}_i + \epsilon_i, \tag{4.7}$$

where pub_rec_i is the number of derogatory records associated with loan i's borrower, and mills_i is the inverse Mills ratio generated by the Heckman correction in Equation (4.6). Note that we do not include time fixed effects for the self-platform analysis. Indeed, the variable after_ref_i equals 1 if loan i was issued after the time event (T_2) and 0 otherwise. Consequently, this variable is highly correlated with the time indicator. If we were to include time fixed effects in the model, it would introduce multicollinearity. Recall that we focus on the coefficient β_3. The results of the regression for P_1 and P_2 are reported in Table 4.7 and are consistent with the t-test results (the full regression tables can be found in the Appendix). The coefficients associated with after_ref_i (β_3) are statistically significant for P_1 in all cases and insignificant for P_2.

Table 4.7: Regression results for P_1 and P_2

Variable	β_3 (standard error) for P_1	β_3 (standard error) for P_2
Default	0.008 (0.002)***	-0.0007 (0.004)
Return	-0.0043 (0.000)***	-0.0011 (0.001)
Credit score	-2.4535 (0.200)***	-0.0442 (0.348)
Annual income	-1232.0369 (232.959)***	-157.3760 (279.638)
Credit history	2.1319 (0.585)***	1.5634 (0.954)

We next incorporate our moderating factor into the second-stage regression model. Extending the specification of Equation (4.7), we obtain:

$$Y_i = \beta_0 + \beta_1 \cdot \text{pub\_rec}_i + \beta_2 \cdot \text{mills}_i + \beta_3 \cdot \text{loan\_type}_i + \beta_4 \cdot \text{loan\_type}_i \times \text{after\_ref}_i + \epsilon_i. \tag{4.8}$$

Compared to Equation (4.7), this specification includes two additional terms: a moderation variable (loan_type_i) and an interaction term ($\text{loan\_type}_i \times \text{after\_ref}_i$).[11] The vector of coefficients β_4 captures the intervention effect on each subgroup of loans. We report the estimated values of β_4 for P_1 and P_2 in Tables 4.8 and 4.9, respectively.

Table 4.8: Regression results for P_1 (with moderator)

Variable	β_4 (standard error)		
	Risky	Average	Safe
Default	0.0268 (0.004)***	0.0073 (0.004)*	0.0019 (0.004)
Return	-0.0096 (0.001)***	-0.0016 (0.001)*	-0.0012 (0.001)
Credit score	1.2996 (0.310)***	-2.1174 (0.313)***	-9.1876 (0.303)***
Annual income	-2716.7904 (397.424)***	-2571.8922 (401.057)***	10.6757 (388.069)
Credit history	-9.1858 (1.009)***	-2.5367 (1.018)	14.9242 (0.985)***

Table 4.9: Regression results for P_2 (with moderator)

Variable	β_4 (standard error)		
	Risky	Average	Safe
Default	0.0008 (0.006)	-0.0061 (0.006)	0.0009 (0.006)
Return	-0.0030 (0.001)**	0.0005 (0.001)	-0.0009 (0.001)
Credit score	-0.4557 (0.522)	-0.1184 (0.520)	1.1776 (0.501)**
Annual income	-598.7015 (488.340)	432.7359 (487.289)	-413.8419 (469.176)
Credit history	2.4049 (1.673)	-1.4939 (1.669)	3.8404 (1.607)**

[11] Note that in Equation (4.8), we dropped the after_ref_i variable so that we can keep all three different loan types (Risky, Average, Safe) as categorical variables (for convenience).

We can see that for P_1, the negative impact of the IPO filing gradually diminishes from Risky to Safe loans. We find that the drop in performance (default rate and return) was most significant for Risky loans. It was less significant for Average loans, whereas for Safe loans the effect was not statistically significant. As for P_2, most variables were insignificant—supporting our argument that P_1's IPO filing had little or no impact on P_2 loans.

4.4.2 Cross-Platform Results

Difference-in-Differences Results

The results presented in the single-platform analysis suggest that the IPO filing event had a negative impact on the performance and quality of P_1 loans but did not affect P_2 loans. In this section, our goal is to study the pre- and post-treatment behavior of loans using P_2 as a control group. We start by fitting a Heckman correction model on the combined dataset:

$$\Pr(A_j = 1) = \Phi(\gamma_0 + \gamma_1 \cdot \text{loan\_amnt}_j + \gamma_2 \cdot \text{dti}_j + \gamma_3 \cdot \text{platform}_j). \tag{4.9}$$

Note that the emp_length$_j$ variable is dropped given that both platforms have different definitions of employment length.[12] We consider four specifications for the second-stage model (with or without monthly time fixed effects and using the original or the matched dataset). We next estimate the diff-in-diffs specification based on Equation (4.4):

$$Y_i = \beta_0 + \beta_1 \cdot \text{pub\_rec}_i + \beta_2 \cdot \text{mills}_i + \beta_3 \cdot \text{time}_i + \beta_4 \cdot \text{intervention}_i + \beta_5 \cdot \text{time\_intervention}_i + \epsilon_i. \tag{4.10}$$

In this case, the coefficient of interest is β_5, which represents the effect of the intervention on the treated group relative to the control group. We also consider including monthly time fixed effects:

$$Y_i = \beta_0 + \beta_1 \cdot \text{pub\_rec}_i + \beta_2 \cdot \text{mills}_i + \beta_3 \cdot \text{month}_i + \beta_4 \cdot \text{intervention}_i + \beta_5 \cdot \text{time\_intervention}_i + \epsilon_i. \tag{4.11}$$

Equation (4.11) is the same as Equation (4.10) except that the variable time$_i$ is replaced by month$_i$. Specifically, time$_i$ is a binary indicator capturing whether loan i was issued before or after the event, whereas month$_i$ is a monthly categorical indicator. Using the categorical indicator allows us to control for monthly time effects. We report the estimated coefficients β_5 for all five DVs in Table 4.10 (the full regression table can be found in the Appendix).

[12] For P_1, emp_length$_j$ is a categorical variable with 11 different levels so that all applicants with emp_length$_j$ > 10 years are assigned to the same category. For P_2, emp_length$_j$ is a numerical variable equal to the months of employment. Given this definition mismatch, including this variable in Equation (4.9) would induce an estimation bias.

Table 4.10: Generic diff-in-diffs results

Variable	β_5 (standard error)	
	w/o time fixed effects	with time fixed effects
Default	0.0068 (0.001)***	0.0047 (0.001)***
Return	-0.0030 (7.23e-05)***	-0.0032 (0.000)***
Credit score	-2.5451 (0.088)***	-1.3457 (0.333)***
Annual income	-270.3765 (240.895)	43.7992 (47.203)
Credit history	0.5879 (0.175)***	3.0874 (0.326)***

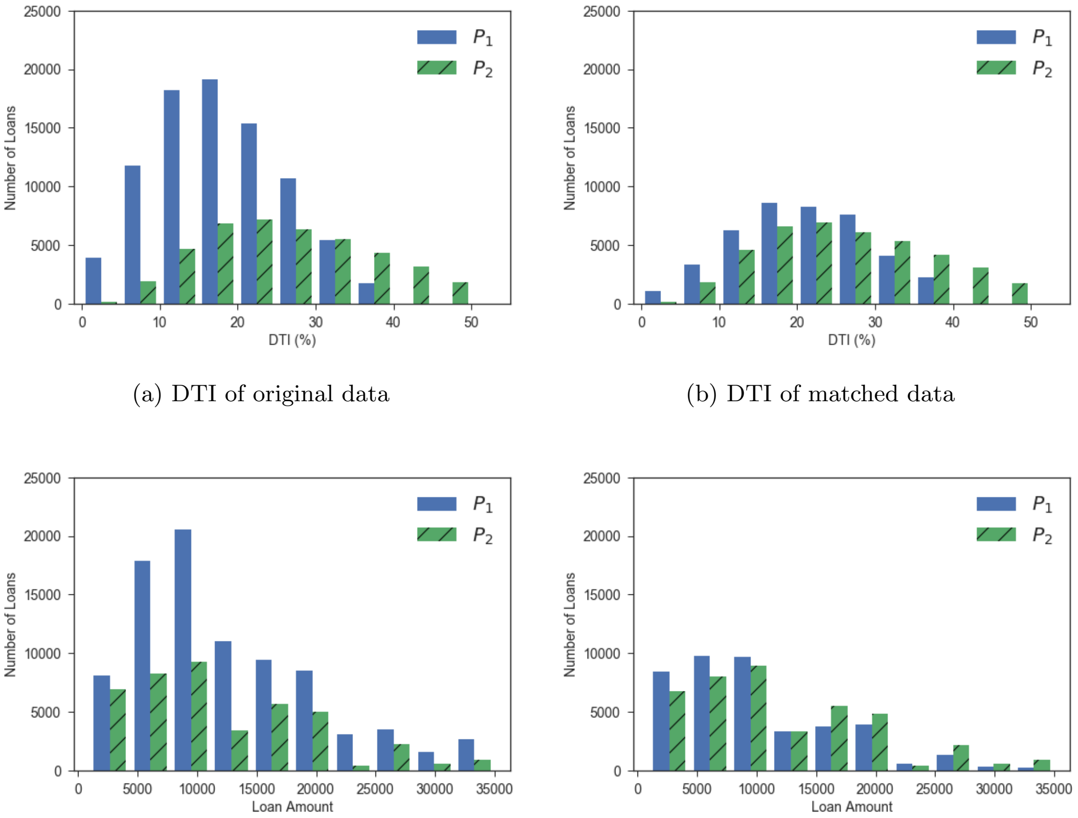
The results are consistent with and without time fixed effects and confirm our findings from the t-tests and regression models. When compared to P_2, P_1 had a significant drop in terms of performance during the post-treatment period. The average credit score decreased, the credit history increased, and the change in annual income was insignificant (in this case). In the single-platform analysis, we observed that the average annual income dropped for P_1. Based on the diff-in-diffs model, however, we can see that the change in annual income for P_1 relative to P_2 was actually not affected by the intervention (we will revisit this result later in this section).

As mentioned in Section 4.3.2, the diff-in-diffs model suffers from the limitation that P_2 may not be considered to be an appropriate control group for P_1. To address this shortcoming, we propose two approaches: estimating the diff-in-diffs model using a subset of carefully selected loans (based on propensity score matching) and the synthetic control method. We next implement the propensity score matching method to select a subsample of P_1 loans which are similar to P_2 loans. Specifically, we estimate the following logit model:

$$PL_i \sim \text{Logit}\left(\beta_0 + \beta_1 \cdot \text{dti}_i + \beta_2 \cdot \text{loan\_amnt}_i\right), \tag{4.12}$$

where PL_i is a binary variable indicating whether loan i belongs to P_1. Note that we use only two explanatory variables (dti_i and loan_amnt_i) in Equation (4.12). This is driven by the fact that other variables, such as credit score and annual income, will be used later as dependent variables. If we were to also use them in the logit model, the matched loans would have values similar to those of the treated loans, hence conflicting with the purpose of our analysis. After obtaining the propensity scores, we implement the nearest-neighbor algorithm to match both datasets (we also implemented caliper matching with width equal to either 0.2 or 0.8 and obtained similar results). To verify that the covariates are balanced after the matching procedure, we present the histograms of both features (DTI and loan amount) in Figure 4.1.

As expected, after running the PSM estimator (see right panels in Figure 4.1), the covariates are much closer than before (left panels). We can then estimate the diff-in-diffs model using this new generated subset of data as the treated group. As before, we consider the model with and without time fixed effects. The results are reported in Table 4.11 and show that after using more balanced data, the results still hold for most variables. The major difference is that the credit history for P_1 becomes lower than that of P_2 (as



(a) DTI of original data

(b) DTI of matched data

(c) Loan amount of original data

(d) Loan amount of matched data

Figure 4.1: Comparison of the histograms using original and matched data

mentioned, credit history is not very indicative anyway).

Table 4.11: Diff-in-diffs results with PSM

Variable	β_5 (standard error)	
	w/o time fixed effect	with time fixed effect
Default	0.0207 (0.002)***	0.0266 (0.000)***
Return	-0.0022 (8.81e-05)***	-0.0014 (0.001)*
Credit score	-4.1802 (0.061)***	-3.4621 (0.144)***
Annual income	-863.4045 (952.279)	-1209.0924 (292.668)***
Credit history	-5.6161 (0.486)***	-5.8326 (2.143)***

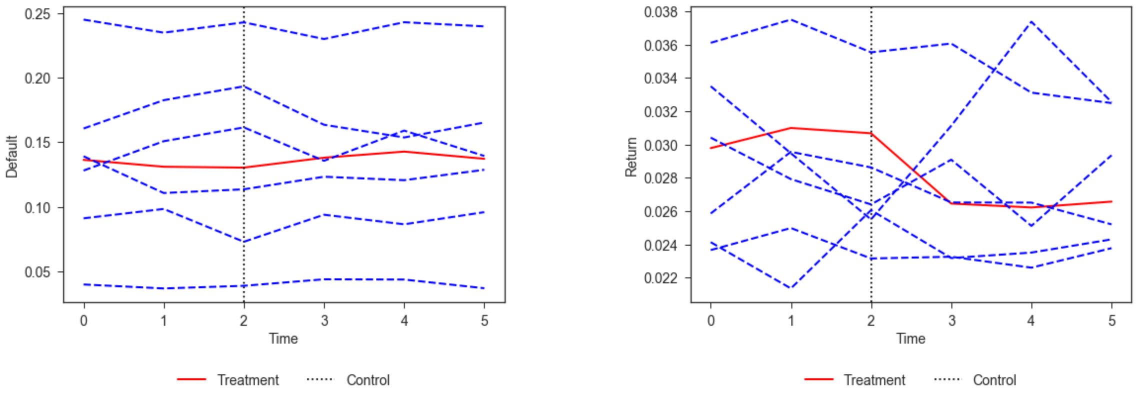
Finally, we complement our analysis by applying our moderating factor to the diff-in-diffs model. The analysis is conducted on both the original and matched datasets. As discussed in Section 4.3.3, the way the moderating factor is applied to the cross-platform analysis is different. Instead of using percentiles, we now use pre-defined thresholds on interest rates to split the loans. Specifically, we use 10% and 15% as threshold values (for robustness purposes, we consider different thresholds in Section 4.6.2). As before, there are two types of specification (with and without time fixed effects), which in total lead to four different models. The regression equation that includes the moderating factor has the following form:

$$\begin{aligned} Y_i &= \beta_0 + \beta_1 \cdot \text{pub\_rec}_i + \beta_2 \cdot \text{mills}_i + \beta_3 \cdot \text{time}_i + \beta_4 \cdot \text{intervention}_i + \\ &+ \beta_5 \cdot \text{loan\_type}_i + \beta_6 \cdot \text{loan\_type}_i \times \text{time\_intervention}_i + \epsilon_i, \end{aligned}$$

where β_3 can either multiply time_i or month_i. As before, to study the moderator effect, we extend the model specification by adding the categorical variable loan_type_i and the interaction term $\text{loan\_type}_i \times \text{time\_intervention}_i$. The results and their interpretations can be found in the Appendix (see Tables 4.22-4.24).

Synthetic Control Results

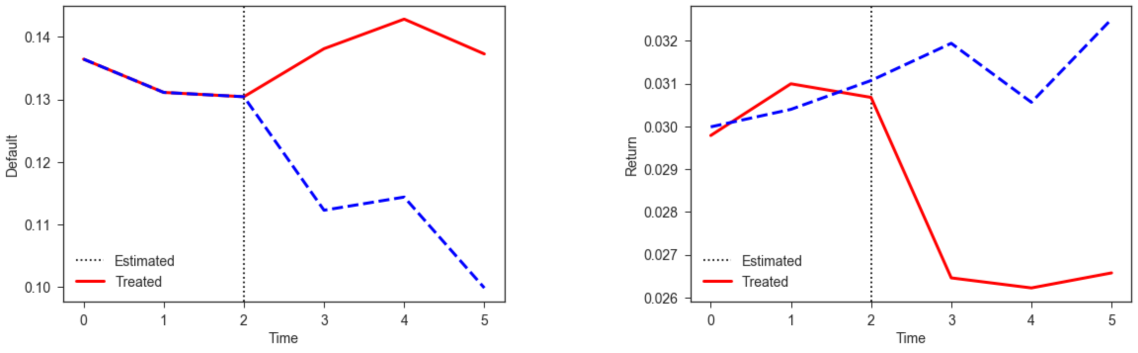
Our previous analyses focused on the relative difference of the dependent variables during pre- and post-treatment periods. In this section, we implement an alternative technique that allows us to visualize the monthly trend throughout the entire period. As mentioned, we create several control groups by splitting the data from P_2 into several disjoint subgroups. The splitting procedure is based on the value of the interest rate or on the loan grade. This allows us to obtain enough variation among the different control groups. Without enough variation, it would be impossible to generate synthetic data that mimic the behavior of the treated group. For example, in Figure 4.2, we plot the variation of two dependent variables (default rate and return) after splitting the data from P_2 into six subgroups (based on interest rate). We then obtain six control curves and one treated curves—allowing us to apply the synthetic control method.



(a) Default

(b) Return

Figure 4.2: Trend visualization of two dependent variables



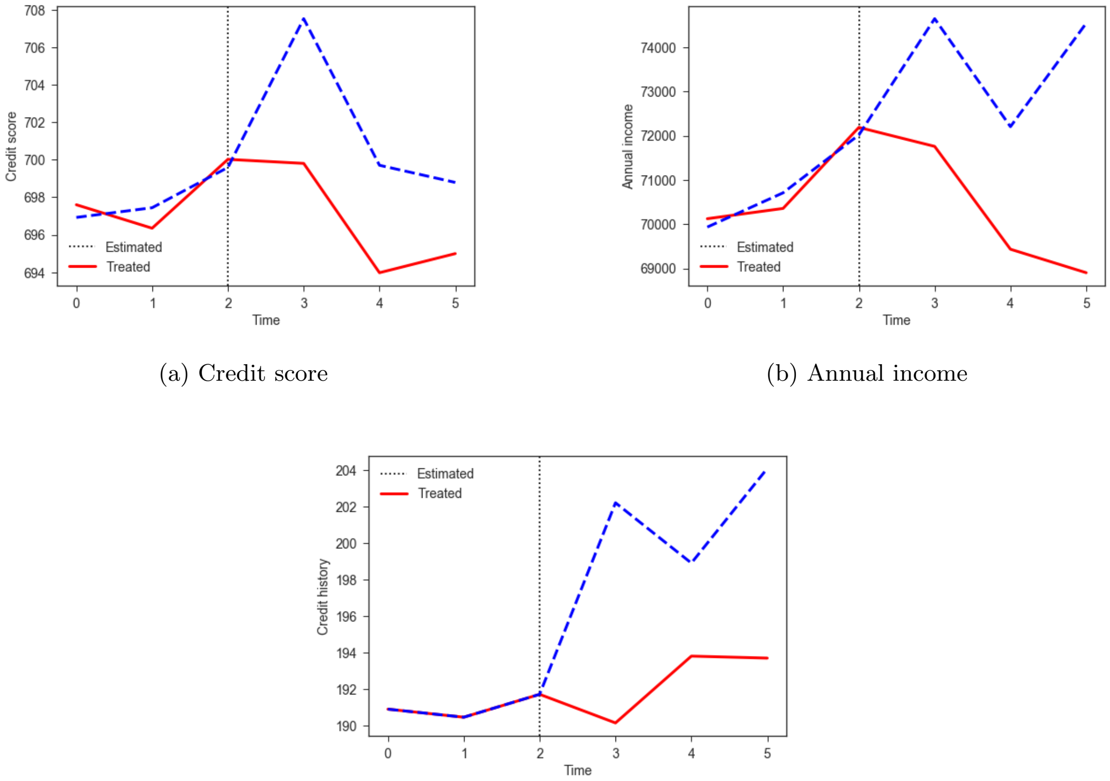
(a) Default

(b) Return

Figure 4.3: Loan performance comparison between pre- and post-treatment

We split the data from P_2 using the following interest-rate cutoffs: 0.08, 0.10, 0.12, 0.14, and 0.16 (we consider a different way of splitting in Section 4.6.4). We then compute the average of each dependent variable for each month and use the result as input to the synthetic control method. Accordingly, we have six control groups (all originate from P_2) and one treated group (P_1). We use a period of three months before and three months after the IPO filing event so that the treatment occurs at $t = 3$ (the pre-treatment months are $t = 1, 2, 3$ and the post-treatment months are $t = 4, 5, 6$). In Figure 4.2, we visualize how the average of two DVs (default rate and return) vary over time for each group (six control and one treated) throughout the entire time period (six months). As we can see, the average default rate of P_1 slightly increased after the event, whereas the return admitted a steep decrease. We next fit the synthetic control method to infer what would have happened if the synthetic data had been treated.

The results are presented in Figure 4.3 (for default and return) and in Figure 4.4 (for credit score, annual income, and credit history). The curves for $t = 1, 2, 3$ represent the trend during the pre-treatment period, and the curves for $t = 4, 5, 6$ correspond to the post-treatment observations. These plots confirm that several



(a) Credit score

(b) Annual income

(c) Credit history

Figure 4.4: Borrowers' attributes comparison between pre- and post-treatment

performance metrics were affected by the IPO filing event. Specifically, the default rate (resp. return) should have dropped (resp. increased) for P_1 after the event based on the estimated trend. However, as shown by the treatment group (solid line), P_1 loans significantly underperformed: the average default rate increased from 10.9% to 13.96%, and the average return decreased from 3.18% to 2.65%. Once again, this suggests that the IPO filing had a negative impact on the overall performance of P_1 loans.

The synthetic control plots for borrowers' attributes are consistent with the diff-in-diffs results. The average credit score and annual income of P_1 decreased more than anticipated after the IPO filing: the average credit score decreased from 702 to 696.3, and the average annual income decreased from 73,750 to 70,083. The result for credit history is the same as the diff-in-diffs result on the matched dataset (using PSM). Although the value increased, the average credit history length is lower when compared to the (synthetic) control group.

Finally, we apply the synthetic control method to the different loan types (Safe, Average, and Risky) separately (the results can be found in the Appendix). For the most part, the results are consistent with our previous findings: the largest performance drop was observed for riskier loans (i.e., loans with a high interest grade or a low grade), whereas the credit score decreased the most for safer loans.

4.4.3 Summary of Results for Other Time Events

So far, our analysis has focused exclusively on T_2 (i.e., the IPO filing event). In this section, we briefly summarize our findings for other time periods (T_1, T_3, and T_4). For each event, we mention both the self-platform and cross-platform results. As a benchmark, we also discuss the results for the year before the IPO, which we refer to as T_0.

Event T_0 – The year before the IPO

- The average return of P_1 loans is stable at approximately 3.7%. The average default rate also remains stable at around 11.5%.
- The average acceptance rate is approximately 12% (although it slightly increases throughout).
- The average credit score of P_1 loans is around 699 and admits a similar behavior.
- We set the reference point in the middle of T_0 (i.e., after six months). We find that there is no significant change in the default rate or in the return for either platform.
- The cross-platform analysis suggests that P_1 slightly outperformed P_2 (i.e., lower default rate and higher return).

Event T_1 – First IPO rumor

- The average default rate of P_1 loans increases from 12.6% to 13.2% (after T_1). The return decreases from 3.4% to 3.1%.
- For P_2, the loan performance also decreases after T_1. The average default rate increases from 13.2% to 14.9%, and the return decreases from 3.7% to 3.1%.
- The cross-platform analysis suggests that P_1 offers higher returns and lower default rates relative to P_2 (the differences are statistically significant).
- The average annual income and credit history of P_1 decreases significantly after T_1 (these variables do not vary for P_2).
- The cross-platform analysis suggests that the average credit score (resp. annual income and credit history) of P_1 is significantly higher (resp. lower) relative to P_2.

As we can see, P_1 generally outperforms P_2 in terms of performance and borrowers' attributes during both T_0 and T_1. However, the results flips when we conduct the same set of comparisons at T_2 (IPO filing).

Event T_3 – Actual IPO date

- The actual IPO does not seem to have a significant impact on P_1 loans in terms of default rate. The default rate remains around 14% before and after T_3. The return decreases from 2.62% to 2.38%.
- The average credit score of P_1 increases by 0.7 points (this increase is statistically significant at the 1% level), and the credit history decreases by one month.
- The acceptance rate of P_1 increases from 12.7% to 14.5%.
- The self-platform results for P_2 show that all borrowers' attributes are inferior after T_3. A potential explanation is that after its IPO, P_1 gained in reputation and became the first choice for borrowers (borrowers might thus go to P_2 only when they were not accepted by P_1).
- The cross-platform analysis suggests that P_1 dominates P_2 across all DVs after T_3.

Overall, the impact of the IPO filing (i.e., T_2) starts to vanish around T_3 so that the metrics are back to their nominal levels (i.e., same as T_0 and T_1).

Event T_4 – Second quarterly report release date

- As before, the average default rate of P_1 loans remains around 14% (the variation is statistically insignificant).
- The average credit score of P_1 increases from 696.2 to 697.5, whereas the credit history decreases from 194.4 to 193.1. Both results are statistically significant.
- The acceptance rate of P_1 is approximately 15.1%.
- For P_2, the results are exactly the same as in T_3, that is, we observe a negative impact across all DVs.
- Similar to T_3, the cross-platform analysis suggests that P_1 has a better performance than P_2 across all DVs we considered.

4.5 Insights and Main Findings

In this chapter, we analyzed four key time events related to P_1's IPO activities. We next summarize and highlight our main findings.

- **Acceptance rate:** The acceptance rate of P_1 keeps increasing throughout all events. Further, the variation is statistically significant at each time event. Although this cannot be directly interpreted as a signal of softening borrowers' requirements, it is still surprising to observe such an increase during the IPO year.

- **Self-platform performance:** Based on the self-platform analysis, both P_1 and P_2 have a slight performance decrease relative to the pre-IPO year (T_0). However, the variation is not statistically significant. Following the IPO filing (T_2), the difference between the two platforms becomes very apparent: P_1 performs worse than before, whereas P_2 remains stable. However, the situation reverses after the IPO (i.e., for T_3 and T_4) as the average default rate of P_1 (resp. P_2) stabilizes (resp. increases). A possible explanation is that after the IPO, P_1's reputation and media attention attracted a large number of favorable applicants.

- **Cross-platform analysis:** Before the IPO-related events, P_1 generally outperforms P_2 in terms of loan performance and borrowers' attributes. However, the situation reverses after T_2 (IPO filing). While P_1 outperforms P_2 during T_0 and T_1, P_1 is dominated by P_2 (in every single dimension we examined) after T_2. Then, the situation suddenly flips back after T_3 so that the negative impact of the IPO filing starts to vanish. Last, P_1 retains its dominance and outperforms P_2 in all dimensions after T_4. This up-down-up trend suggests that P_1 may have modified some of its operations after its IPO filing.

- **Different loan types:** Safe loans (i.e., low interest rate) are generally stable in both platforms. However, the average credit score of P_1's Safe loans drop by more than 10 points after T_2 (IPO filing). This decrease is substantial given that we could not find a comparable drop at any other time period in our dataset. Following T_3, P_1's Safe loans become superior again. As for Risky loans, P_1 usually outperforms P_2 except for T_2, during which P_1 is actually worse than P_2. These results raise the question to as whether P_1 adjusted its screening process after its IPO filing.

In summary, our results suggest that during its IPO year, P_1 may have softened its requirements and accepted lower-performing loans. The IPO filing event (T_2) had a significant negative impact on the platform performance and on borrowers' attributes. After the actual IPO date, the negative effect gradually dissipated to retrieve the initial status where P_1 outperforms P_2.

4.6 Robustness

In this section, we present robustness checks (for T_2) along four dimensions: we vary the number of months included in the analysis, we test an alternative definition of our moderating factor, we consider alternative return definitions, and we use a different split for synthetic control.

4.6.1 Varying the Number of Months

In Section 4.4, we analyzed the treatment effect by looking at loans issued three months before and three months after a specific event. We denote this value by Δt. In this section, we set $\Delta t = 4$.

Table 4.12: Generic diff-in-diffs results ($\Delta t = 4$)

Variable	β_5 (standard error)	
	w/o time fixed effects	with time fixed effects
Default	0.0034 (0.001)***	0.0013 (0.001)*
Return	-0.0021 (4.46e-05)***	-0.0019 (2.2e-05)***
Credit score	-1.4293 (0.020)***	-1.3999 (0.163)***
Annual income	-175.0383 (204.942)	-164.1894 (40.405)
Credit history	2.5540 (0.153)***	4.0522 (0.216)***

The diff-in-diffs results (using the original datasets) for $\Delta t = 4$ are presented in Table 4.12. When using $\Delta t = 4$, our findings remain consistent with $\Delta t = 3$ (the t-test and regression results also remain similar and are omitted for conciseness). Although the change in default rate becomes less significant (when including time fixed effects), P_1 still seems to underperform P_2.

We only consider $\Delta t = 3$ and $\Delta t = 4$. Indeed, when $\Delta t = 1, 2$, there are not enough observations to ensure statistical power. If we were to use $\Delta t > 4$, the samples would overlap with other time events (T_1 and T_3) as the distance between two consecutive events is approximately four months.

4.6.2 Alternative Risk Level Definition

In Section 4.3, we used the following return definition when estimating our diff-in-diffs specification:

$$\text{Risk Level}_i = \begin{cases} \text{Safe} & \text{if Int Rate}_i < r_1, \\ \text{Medium} & \text{if } r_1 \leq \text{Int Rate}_i < r_2, \\ \text{Risky} & \text{if Int Rate}_i \geq r_2, \end{cases}$$

where $r_1 = 0.1$ and $r_2 = 0.15$. In this section, we examine different threshold values and check if our results still hold. Specifically, we vary the values of (r_1, r_2). We first set $(r_1, r_2) = (0.09, 0.16)$ and report the results in Tables 4.13-4.15.

Table 4.13: Diff-in-diffs results for Risky loans ($r_1 = 0.09, r_2 = 0.16$)

Variable	β_6 (standard error)			
	Model 1	Model 2	Model 3	Model 4
Default	0.0465 (0.001)***	0.0425 (0.001)***	0.0295 (0.015)**	0.0279 (0.012)**
Return	-0.0145 (0.001)***	-0.0146 (0.001)***	-0.0016 (0.000)**	-0.0016 (0.000)***
Credit score	6.2510 (3.880)	7.4355 (3.367)**	6.4638 (4.211)	7.5213 (3.838)*
Annual income	-4909.8358 (2368.608)**	-4572.0647 (2707.630)*	-4912.3536 (135.746)***	-4820.1811 (1270.639)***
Credit history	-11.6499 (0.249)***	-8.6620 (0.686)***	-13.5445 (4.031)***	-12.0222 (5.773)**

Table 4.14: Diff-in-diffs results for Average loans ($r_1 = 0.09, r_2 = 0.16$)

Variable	β_6 (standard error)			
	Model 1	Model 2	Model 3	Model 4
Default	0.0153 (1.4e-05)***	0.0111 (0.000)***	0.0343 (0.004)***	0.0333 (0.007)***
Return	-0.0007 (0.001)	-0.0008 (0.000)	-0.0021 (0.000)***	-0.0017 (0.001)**
Credit score	-3.9734 (1.131)***	-2.6573 (1.642)	-4.8558 (1.342)***	-3.2964 (1.756)*
Annual income	-952.9084 (393.835)**	-537.9750 (123.132)***	81.6259 (1739.092)	-16.0799 (257.884)
Credit history	-4.8943 (0.020)***	-1.9985 (0.543)***	-5.6530 (2.312)	-4.6691 (0.229)***

Table 4.15: Diff-in-diffs results for Safe loans ($r_1 = 0.09, r_2 = 0.16$)

Variable	β_6 (standard error)			
	Model 1	Model 2	Model 3	Model 4
Default	0.0034 (0.002)*	-0.0009 (0.001)	0.0037 (0.003)	0.0031 (0.001)***
Return	0.0002 (0.001)	7.459e-05 (0.001)	-0.0007 (0.000)	-2.824e-05 (0.000)
Credit score	-16.2659 (0.603)***	-14.9146 (1.137)***	-20.4647 (2.614)***	-18.5880 (3.126)***
Annual income	1789.7595 (2442.571)	2283.6045 (2096.533)	-1288.9554 (135.746)***	-1520.7169 (1546.236)
Credit history	13.6711 (1.918)***	16.7099 (1.453)***	-4.1079 (0.110)***	-3.2358 (0.210)

As we can see, most of the statistically significant results remain. In the Appendix, we present the results for $(r_1, r_2) = (0.11, 0.14)$ (similar results were also obtained with $(r_1, r_2) = (0.08, 0.14)$ and $(r_1, r_2) = (0.11, 0.16)$). Under these threshold values, we observe the same patterns as before: (i) Risky loans of P_1 underperformed relative to P_2 loans, (ii) the average annual income of Risky loans decreased after the IPO filing, and (iii) for Average and Safe loans, attributes such as credit score and credit history decreased after the IPO filing.

4.6.3 Alternative Return Definitions

For robustness purposes, we consider two alternative return definitions, as introduced in [24] in the context of peer-to-peer loans:

- Method 2 (M2): If the loan is fully paid, the investor can immediately re-invest the capital with the exact same return as before. If the loan defaults, the investor will suffer the loss and not re-invest (same as M1). We thus have the following formula:

$$\begin{cases} \frac{p-f}{f} \cdot \frac{12}{m} & \text{if } p - f > 0, \\ \frac{p-f}{f} \cdot \frac{12}{t} & \text{if } p - f \leq 0, \end{cases}$$

  where m is the actual loan duration in months, that is, the time that the loan actually lasted before reaching the status fully paid or defaulted.

- Method 3 (M3): Consider a total investment horizon of T months. Assume that the payments were made uniformly throughout the loan duration (m months). The monthly cashflow would then be of size p/m. Assume also that this capital would be re-invested at a risk-free rate i. We can use the sum of a geometric series to compute the total return:

$$\frac{12}{T} \cdot \frac{1}{f} \quad \frac{p}{m} \, \frac{1-(1+i)^m}{1-(1+i)} \quad (1+i)^{T-m} - f \quad .$$

We use a time horizon of $T = 36$ months and an annual risk-free rate of $i = 3\%$. The results are reported in Tables 4.16 and 4.17.

Table 4.16: T-test results for P_1 under different return definitions

Return definition	Value before	Value after	P-value (significance)
M1	3.06%	2.63%	4.54e-19***
M2	5.81%	5.31%	2.20e-19***
M3	3.75%	3.31%	4.20e-19***

Table 4.17: T-test results for P_2 under different return definitions

Return definition	Value before	Value after	P-value (significance)
M1	2.97%	2.87%	0.2382
M2	4.40%	4.42%	0.1871
M3	3.49%	3.43%	0.4660

Under all three return definitions, the return of P_1 dropped significantly after the IPO filing event (i.e., T_2), whereas the return of P_2 did not vary significantly. To complement Table 4.10, we also present the diff-in-diffs results under all three return definitions (see Table 4.18).

Table 4.18: Diff-in-diffs results under different return definitions

Return definition	β_5 (standard error)	
	w/o time fixed effects	with time fixed effects
M1	-0.0030 (7.23e-05)***	-0.0032 (0.000)***
M2	-0.0060 (9.02e-05)***	-0.0054 (0.000)***
M3	-0.0054 (8.89e-05)***	-0.0052 (0.000)***

Once again, we retrieve the result that the average return of P_1 loans is negatively affected by the IPO filing event.

4.6.4 Alternative Split for Synthetic Control

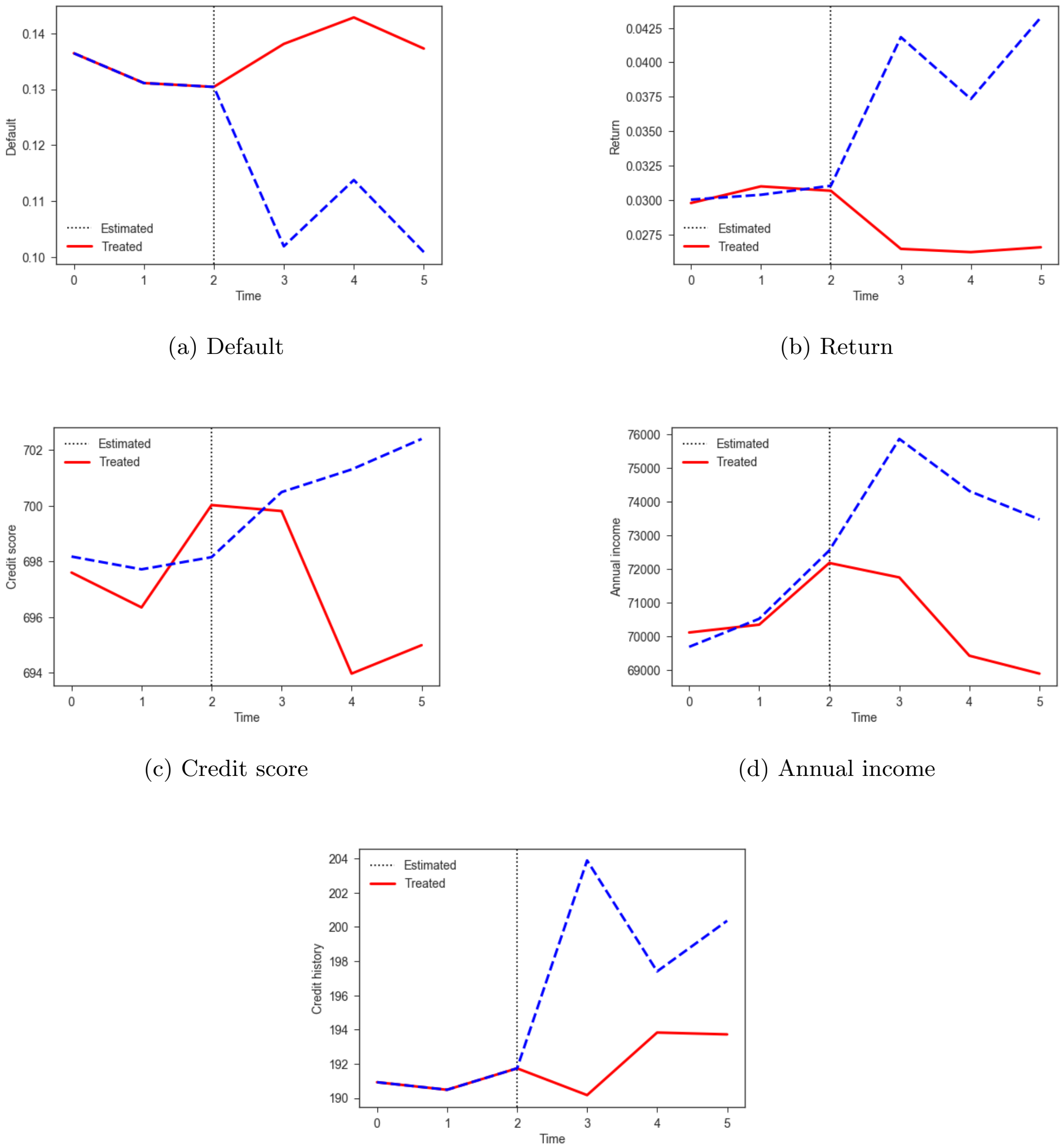
As mentioned, the synthetic control method requires multiple control groups. In our case, given that we only have a single potential control group, we proposed to split it into subgroups. In this section, we check the robustness of our results with respect to the way of splitting our control group (P_2). Specifically, instead of splitting the data from P_2 based on interest rate, we split the loans according to their grade. In this case, each control group simply corresponds to a specific grade (this procedure still creates six control groups as P_2 uses six different grades). Based on this splitting procedure, Figure 4.5 presents the revised synthetic control plots. As we can see, the plots are similar to the plots presented in Section 4.4.2.

4.7 Conclusion

In this chapter, we used a large dataset from two peer-to-peer lending platforms (denoted P_1 and P_2). While P_1 went public by filing an IPO, P_2 remained privately held. We exploit this empirical environment to carefully infer the causal effect of IPOs on peer-to-peer lending platforms. In particular, our analysis aims to investigate whether P_1 altered its decisions in anticipation of its IPO and to quantify the impact of the IPO on investors and on borrowers.

We presented rigorous empirical analyses at four time events: first IPO rumor, filing date, actual IPO, and quarterly report release. For each event, we estimated several econometric models and tested the robustness of our results. Given that our data include both accepted and rejected loans, we proposed to estimate a two-stage regression model to correct for the sample selection bias. By looking at the data from each platform separately, we found that the IPO filing event had a significant impact on P_1 loans but not on P_2 loans. Specifically, P_1 loans' performance (default rate and return) significantly decreased, and borrowers' attributes (credit score and annual income) diminished. We next used the data from P_2 as a control group to conduct cross-platform analyses. Given that P_1 and P_2 are not perfectly comparable, we used the PSM method to select a subset of loans that share similar features across both platforms. We then estimated a diff-in-diffs model using the PSM samples. We also implemented the synthetic control method as an additional approach to validate our results.

We found that several performance metrics were indeed affected by the IPO, especially around the IPO filing date. When comparing loans before and after the IPO filing (while using P_2 loans as a control), we observed that (i) the loans' performance significantly decreased, (ii) borrowers' requirements diminished, and (iii) the acceptance rate of P_1 inflated. These findings suggest that P_1 may have altered some of its decisions in anticipation of its IPO by lowering its requirements for accepting loans. Namely, it is plausible that P_1 had issued loans with lower credit scores and annual incomes right after its IPO filing. Interestingly, after the actual IPO date, all performance metrics returned to their initial levels. The research presented in this chapter has two main implications. First, it seems that P_1 decided to prioritize short-term revenues over long-term reputation due to the desire to boost its predicted market capital value. Second, when peer-to-



(a) Default

(b) Return

(c) Credit score

(d) Annual income

(e) Credit history

Figure 4.5: Comparison of pre- and post-treatment loans

peer lending platforms (or other peer-to-peer financial services) announce their IPO, it may have important implications on both sides of the market (investors and borrowers). In light of the prospective IPOs of several unicorn sharing economy platforms (e.g., Uber, Airbnb), such implications will become increasingly important.

4.8 Appendix

4.8.1 Full Regression Tables

We next report the full tables for the regression model in Equation (4.7) and for the diff-in-diffs model in Equation (4.11). Note that for P_1, the coefficients of *after_ref* (i.e., β_3 in Equation (4.7)) are statistically significant across all three dependent variables, whereas for P_2 they are not statistically significant.

Table 4.19: Estimated regression coefficients of P_1 for default rate, return, and credit score

	DV: Default	DV: Return	DV: Credit score
Intercept	0.1242 (0.002)***	0.0322 (0.000)***	699.6462 (0.203)***
pub_rec	0.0128 (0.002)***	0.0005 (0.000)	-10.3979 (0.171)***
mills	-0.0062 (0.002)***	0.0020 (0.000)***	-0.3903 (0.156)**
after_ref	0.0080 (0.002)***	-0.0043 (0.000)***	-2.4535 (0.200)***
R^2	0.001	0.001	0.043
Number of observations		86,223	

Table 4.20: Estimated regression coefficients of P_2 for default rate, return, and credit score

	DV: Default	DV: Return	DV: Credit score
Intercept	0.3569 (0.017)***	0.0571 (0.004)***	650.1167 (1.627)***
pub_rec	0.0144 (0.003)***	0.0016 (0.001)**	-17.5239 (0.260)***
mills	0.3754 (0.029)***	0.0495 (0.000)***	-112.3444 (2.857)***
after_ref	-0.0007 (0.004)	-0.0011 (0.001)	-0.0442 (0.348)
R^2	0.005	0.001	0.043
Number of observations		41,416	

Table 4.21: Estimated diff-in-diffs coefficients for default rate, return, and credit score

	DV: Default	DV: Return	DV: Credit score
Intercept	-0.1768 (0.067)***	0.0415 (0.006)***	742.5860 (8.549)***
Month [6]	0.0065 (0.001)***	-0.0013 (0.001)*	-0.5237 (0.019)***
Month [7]	0.0021 (0.001)*	-0.0007 (0.000)***	-1.2676 (0.881)
Month [8]	-0.0009 (0.000)**	-0.0010 (0.000)***	1.3914 (0.699)**
Month [9]	0.0014 (0.000)***	-0.0019 (1.07e-05)***	1.6959 (1.111)
Month [10]	0.0054 (0.001)***	-0.0021 (4.6e-05)***	-1.7986 (0.926)*
Month [11]	0.0001 (0.001)	-0.0018 (3.91e-05)***	-1.2728 (0.503)**
pub_rec	0.0154 (0.004)***	0.0011 (0.001)	-13.5430 (3.871)***
mills	-0.5485 (0.112)***	0.0191 (0.009)**	52.3330 (16.655)***
intervention	-0.6892 (0.137)***	0.0245 (0.011)**	53.2558 (20.986)**
time_intervention	0.0047 (0.001)***	-0.0032 (0.000)***	-1.3457 (0.333)***
R^2	0.008	0.001	0.096
Number of observations	127,639		

4.8.2 Diff-in-Diffs Results with Moderating Factor

Table 4.22: Diff-in-diffs results for Risky loans

Variable	β_6 (standard error)			
	Model 1	Model 2	Model 3	Model 4
Default	0.0232 (0.003)***	0.0208 (0.003)***	0.0557 (0.017)***	0.0552 (0.012)***
Return	-0.0096 (0.001)***	-0.0097 (0.001)***	-0.0097 (0.000)***	-0.0089 (0.000)***
Credit score	4.1591 (3.215)	5.2533 (2.767)*	4.4886 (3.855)	5.4669 (3.658)
Annual income	-2830.7341 (1533.168)*	-2558.6206 (1814.226)	-3188.3145 (276.468)***	-3336.9430 (945.923)***
Credit history	-9.1339 (1.798)***	-6.6586 (2.369)***	-6.4028 (1.835)***	-5.4826 (3.667)*

Table 4.23: Diff-in-diffs results for Average loans

Variable	β_6 (standard error)			
	Model 1	Model 2	Model 3	Model 4
Default	0.0083 (0.002)***	0.0061 (0.002)***	-0.0002 (0.014)	-0.0010 (0.018)
Return	0.0007 (0.001)	0.0006 (0.0001)	0.0021 (0.001)	0.0033 (0.002)**
Credit score	-3.4056 (1.289)***	-2.2968 (1.747)	-2.1486 (0.943)	0.9441 (1.324)
Annual income	-969.9138 (380.386)**	-674.8721 (130.338)***	239.1037 (1006.072)	258.0044 (650.798)
Credit history	-4.7383 (1.028)***	-2.3312 (0.447)***	-7.4556 (2.964)*	-7.3711 (0.795)***

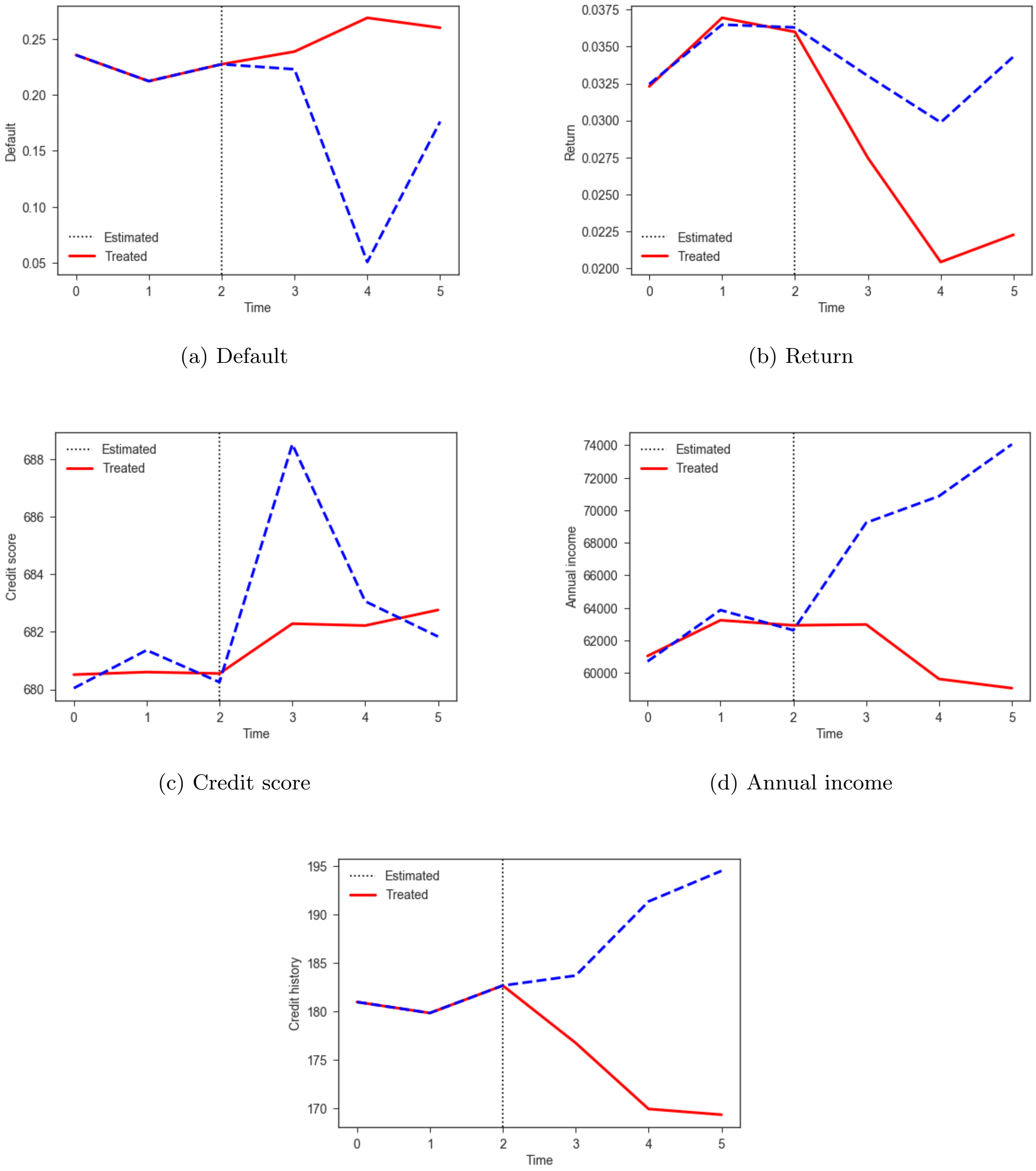
Table 4.24: Diff-in-diffs results for Safe loans

Variable	β_6 (standard error)			
	Model 1	Model 2	Model 3	Model 4
Default	0.004 (0.001)***	0.0016 (0.001)	0.0025 (0.001)***	0.0033 (0.003)
Return	0.0007 (0.001)	0.0005 (0.0001)	0.0019 (0.001)	0.0031 (0.000)***
Credit score	-12.0502 (2.455)***	-10.9624 (2.912)***	-16.6716 (4.736)***	-15.5057 (5.033)***
Annual income	1971.4810 (2121.071)	2277.5110 (1819.404)	707.8679 (1733.951)	498.2058 (235.120)**
Credit history	11.7184 (1.641)***	14.2248 (1.110)***	-2.5756 (0.169)***	-2.4277 (3.062)

The definitions of all four models are as follows: Models 1 and 2 use the original data without and with time fixed effects, respectively, whereas Models 3 and 4 use the matched data without and with time fixed effects, respectively.

There are several interesting observations that confirm our findings from the self-platform analysis. First, for Risky loans, the t-test and regression results suggested that P_1 had a significant drop in both performance and attributes after the event, while P_2 did not show any significant change. Based on Table 4.22, we can see that the decrease in performance metrics and credit history length are statistically significant across all four models (the decrease in annual income is significant across three models out of four). This provides a good indication that P_1 was worse off than P_2 after T_2. For Average and Safe loans, we can see from Tables 4.23 and 4.24 that the credit history length of P_1 Average loans and the credit score of P_1 Safe loans are much worse relative to P_2—confirming our previous results. There are very few instances where P_1 loans were better than P_2 loans during this time period. As we saw in Section 4.4.3, however, P_1 loans typically outperformed P_2 loans for these DVs during other time periods.

4.8.3 Synthetic Control Results for Risky, Average, and Safe Loans



(a) Default

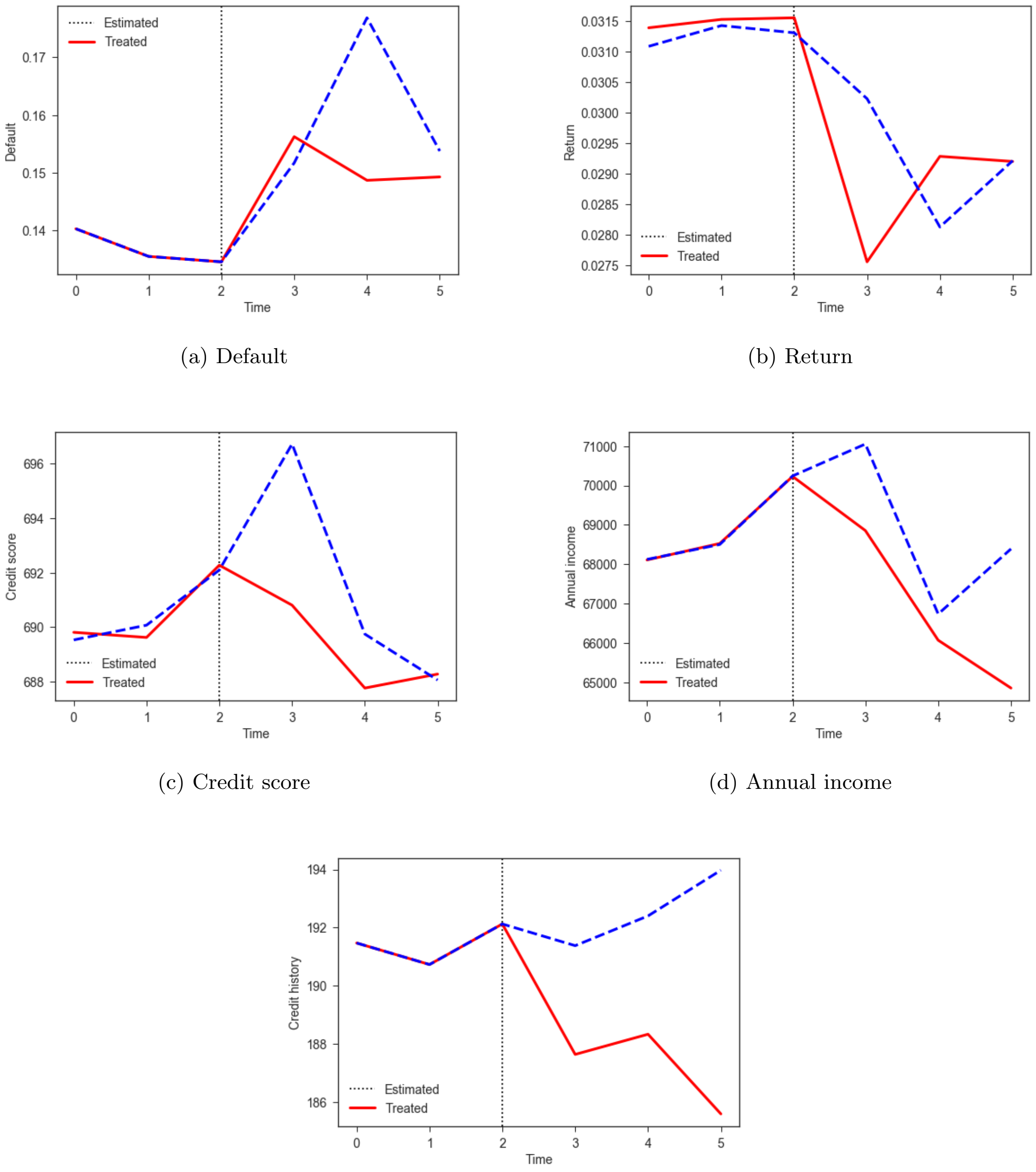
(b) Return

(c) Credit score

(d) Annual income

(e) Credit history

Figure 4.6: Comparison of pre- and post-treatment for Risky loans



(a) Default

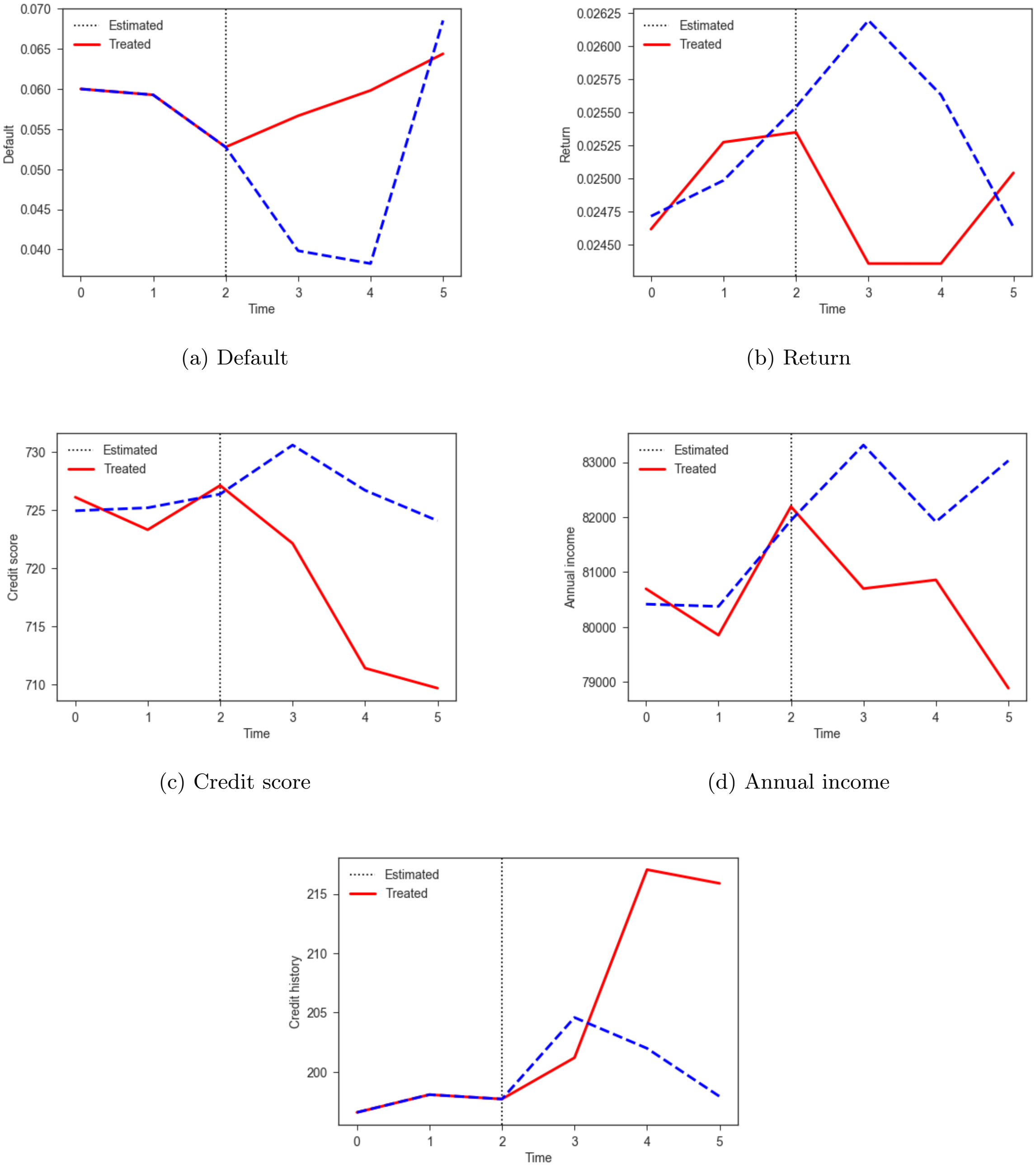
(b) Return

(c) Credit score

(d) Annual income

(e) Credit history

Figure 4.7: Comparison of pre- and post-treatment for Average loans



(a) Default

(b) Return

(c) Credit score

(d) Annual income

(e) Credit history

Figure 4.8: Comparison of pre- and post-treatment for Safe loans

4.8.4 Diff-in-Diffs Results for $\mathbf{r_1 = 0.11}$ and $\mathbf{r_2 = 0.14}$

Table 4.25: Diff-in-diffs results for Risky loans ($r_1 = 0.11, r_2 = 0.14$)

Variable	β_6 (standard error)			
	Model 1	Model 2	Model 3	Model 4
Default	0.0267 (0.000)***	0.0244 (0.000)***	0.0468 (0.017)*	0.0472 (0.014)***
Return	-0.0102 (0.002)***	-0.0103 (0.002)***	-0.0064 (0.001)***	-0.0058 (0.001)***
Credit score	5.7296 (4.034)	6.8292 (3.602)*	5.9759 (3.958)	6.9396 (3.771)*
Annual income	-2950.9351 (1329.923)**	-2677.1608 (1615.140)*	-1588.7019 (468.454)***	-1887.6066 (595.762)***
Credit history	-9.0029 (1.775)***	-6.5352 (2.365)***	-7.2363 (3.915)*	-5.9330 (6.245)

Table 4.26: Diff-in-diffs results for Average loans ($r_1 = 0.11, r_2 = 0.14$)

Variable	β_6 (standard error)			
	Model 1	Model 2	Model 3	Model 4
Default	0.0120 (0.002)***	0.0098 (0.002)***	0.0185 (0.018)	0.0183 (0.021)
Return	-0.0005 (0.001)	-0.0006 (0.001)	-0.0006 (0.002)	0.0002 (0.002)
Credit score	-2.7447 (1.121)**	-1.6295 (1.557)	-2.4275 (0.341)***	-1.1862 (0.685)*
Annual income	-1315.5111 (73.318)***	-1029.2349 (160.253)***	-2178.4788 (1320.489)*	-2135.2684 (309.370)
Credit history	-7.7932 (1.228)***	-5.4366 (0.616)***	-11.9193 (6.021)**	-12.1191 (2.232)***

Table 4.27: Diff-in-diffs results for Safe loans ($r_1 = 0.11, r_2 = 0.14$)

Variable	β_6 (standard error)			
	Model 1	Model 2	Model 3	Model 4
Default	-0.0013 (0.002)	-0.0036 (0.002)	0.0048 (0.005)	0.0062 (0.000)***
Return	0.0003 (0.000)	0.0001 (0.000)	-2.164e-05 (0.001)	0.0009 (0.001)
Credit score	-10.0258 (2.289)***	-8.9311 (2.726)***	-14.6335 (4.026)***	-13.4088 (4.343)***
Annual income	1847.3566 (1582.506)	2153.1128 (1288.370)*	860.0590 (1120.472)	578.5387 (340.000)*
Credit history	11.4619 (0.926)***	13.9452 (0.368)***	1.1244 (0.140)***	1.4308 (2.652)

Bibliography

[1] Alberto Abadie, Alexis Diamond, and Jens Hainmueller. Synthetic control methods for comparative case studies: Estimating the effect of californias tobacco control program. *Journal of the American Statistical Association*, 105(490):493–505, 2010.

[2] Sreangsu Acharyya. *Learning to rank in supervised and unsupervised settings using convexity and monotonicity*. PhD thesis, Electrical and Computer Engineering, The University of Texas at Austin, 2013.

[3] Rajesh K Aggarwal, Amiyatosh K Purnanandam, and Guojun Wu. Underwriter manipulation in initial public offerings. Working paper, 2005.

[4] Nir Ailon. An active learning algorithm for ranking from pairwise preferences with an almost optimal query complexity. *Journal of Machine Learning Research*, 13(1):137–164, 2012.

[5] Franklin Allen and Gerald R Faulhaber. Signalling by underpricing in the ipo market. *Journal of Financial Economics*, 23(2):303–323, 1989.

[6] Lennart Baardman, Igor Levin, Georgia Perakis, and Divya Singhvi. Leveraging comparables for new product sales forecasting. Available at SSRN 3086237, 2017.

[7] Yoram Bachrach, Tom Minka, John Guiver, and Thore Graepel. How to grade a test without knowing the answers - a Bayesian graphical model for adaptive crowdsourcing and aptitude testing. In *International Conference on Machine Learning (ICML)*, 2012.

[8] M. J. Beal. *Variational Algorithms for Approximate Bayesian Inference*. PhD thesis, Gatsby Computational Neuroscience Unit, University College London, 2003.

[9] Shai Bernstein. Does going public affect innovation? *The Journal of Finance*, 70(4):1365–1403, 2015.

[10] Dimitris Bertsimas and Nathan Kallus. From predictive to prescriptive analytics. arXiv preprint arXiv:1402.5481, 2014.

[11] Richard Blundell, Thomas MaCurdy, and Costas Meghir. Labor supply models: Unobserved heterogeneity, nonparticipation and dynamics. *Handbook of Econometrics*, 6:4667–4775, 2007.

[12] Andrew P Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159, 1997.

[13] R. A. Bradley and M. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39:324345, 1952.

[14] Mark Braverman and Elchanan Mossel. Noisy sorting without resampling. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2008.

[15] Stephen J Brown, William Goetzmann, Roger G Ibbotson, and Stephen A Ross. Survivorship bias in performance studies. *The Review of Financial Studies*, 5(4):553–580, 1992.

[16] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *International Conference on Machine Learning (ICML)*, 2005.

[17] Yunbo Cao, Jun Xu, Tie-Yan Liu, Hang Li, Yalou Huang, and Hsiao-Wuen Hon. Adapting ranking svm to document retrieval. In *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2006.

[18] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *International Conference on Machine Learning (ICML)*, 2007.

[19] Richard B Carter, Frederick H Dark, and Ajai K Singh. Underwriter reputation, initial returns, and the long-run performance of ipo stocks. *The Journal of Finance*, 53(1):285–311, 1998.

[20] Xi Chen, Paul N. Bennett, Kevyn Collins-Thompson, and Eric Horvitz. Pairwise ranking aggregation in a crowdsourced setting. In *ACM International Conference on Web Search and Data Mining (WSDM)*, 2013.

[21] Xi Chen, Kevin Jiao, and Qihang Lin. Bayesian decision process for cost-efficient dynamic ranking via crowdsourcing. *The Journal of Machine Learning Research*, 17(1):7617–7656, 2016.

[22] Xi Chen, Qihang Lin, and Dengyong Zhou. Statistical decision making for optimal budget allocation in crowd labelling. *Journal of Machine Learning Research*, 16:1–46, 2015.

[23] Maxime Cohen and Kevin Jiao. The impact of ipo on peer-to-peer lending platforms. *Available at SSRN*, 2018.

[24] Maxime C Cohen, C Daniel Guetta, Kevin Jiao, and Foster Provost. Data-driven investment strategies for peer-to-peer lending: A case study for teaching data science. *Big Data*, 6(3):191–213, 2018.

[25] Maxime C Cohen, Ngai-Hang Zachary Leung, Kiran Panchamgam, Georgia Perakis, and Anthony Smith. The impact of linear optimization on promotion planning. *Operations Research*, 65(2):446–468, 2017.

[26] K. Collins-Thompson and J. Callan. A language modeling approach to predicting reading difficulty. In *HLT*, 2004.

[27] Lin William Cong and Sabrina T Howell. Listing delays and innovation: Evidence from chinese ipos. Technical report, National Bureau of Economic Research, 2018.

[28] Lee G Cooper, Penny Baron, Wayne Levy, Michael Swisher, and Paris Gogos. Promocast: A new forecasting method for promotion planning. *Marketing Science*, 18(3):301–316, 1999.

[29] William S. Cooper, Fredric C. Gey, and Daniel P. Dabney. Probabilistic retrieval based on staged logistic regression. In *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1992.

[30] Koby Crammer and Yoram Singer. Pranking with ranking. In *Advances in Neural Information Processing Systems (NIPS)*, 2001.

[31] Masako Darrough and Srinivasan Rangan. Do insiders manipulate earnings when they sell their shares in an initial public offering? *Journal of accounting research*, 43(1):1–33, 2005.

[32] A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society Series C*, 28:20–28, 1979.

[33] Priya Donti, Brandon Amos, and J Zico Kolter. Task-based end-to-end model learning in stochastic optimization. In *Advances in Neural Information Processing Systems*, pages 5484–5494, 2017.

[34] Adam N Elmachtoub and Paul Grigas. Smart" predict, then optimize". arXiv preprint arXiv:1710.08005, 2017.

[35] Seyda Ertekin, Haym Hirsh, and Cynthia Rudin. Wisely using a budget for crowdsourcing. Technical report, MIT, 2012.

[36] Ludwig Fahrmeir, Heinz Kaufmann, et al. Consistency and asymptotic normality of the maximum likelihood estimator in generalized linear models. *The Annals of Statistics*, 13(1):342–368, 1985.

[37] Robert Fildes, Paul Goodwin, and Dilek Önkal. Use and misuse of information in supply chain forecasting of promotion effects. *International Journal of Forecasting*, 35(1):144–156, 2019.

[38] Peter C. Fishburn. Binary choice probabilities: on the varieties of stochastic transitivity. *Journal of Mathematical Psychology*, 10(4):327 – 352, 1973.

[39] Eijte W Foekens, Peter SH Leeflang, and Dick R Wittink. Varying parameter models to accommodate dynamic promotion effects. *Journal of Econometrics*, 89(1-2):249–268, 1998.

[40] P. Frazier, W. B. Powell, and S. Dayanik. A knowledge-gradient policy for sequential information collection. *SIAM Journal on Control and Optimization*, 47(5):2410–2439, 2008.

[41] Peter Frazier. *Knowledge-Gradient Methods for Statistical Learning*. PhD thesis, Princeton University, 2009.

[42] Yoav Freund, Raj Iyer, Robert E. Schapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4(11):933–969, 2003.

[43] John M Friedlan. Accounting choices of issuers of initial public offerings. *Contemporary accounting research*, 11(1):1–31, 1994.

[44] C. Gao and D. Zhou. Minimax optimal convergence rates for estimating ground truth from crowd-sourced labels. arXiv:1310.5764, 2013.

[45] Victor F Gauto et al. An econometric analysis of trade creation and trade diversion in mercosur: the case of paraguay. In *International Association of Agricultural Economists Triennial Conference*, pages 1–26, 2012.

[46] David Gleich and Lek heng Lim. Rank aggregation via nuclear norm minimization. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2011.

[47] William H Greene. *Econometric analysis*. Pearson Education India, 2003.

[48] M. Grötschel, M. Jünger, and G. Reinelt. A cutting plane algorithm for the linear ordering problem. *Operations Research*, 32(6):1195–1220, 1984.

[49] S. S. Gupta and K. J. Miescke. Bayesian look ahead one-stage sampling allocations for selection of the best population. *Journal of Statistical Planning and Inference*, 54(2):229–244, 1996.

[50] James J Heckman. Sample selection bias as a specification error (with an application to the estimation of labor supply functions), 1977.

[51] Kevin B Hendricks, Manpreet Hora, and Vinod R Singhal. An empirical investigation on the appointments of supply chain and operations management executives. *Management Science*, 61(7):1562–1583, 2014.

[52] R. Herbrich, T. Minka, and T. Graepel. Trueskill (TM): a bayesian skill rating system. In *Advances in Neural Information Processing Systems (NIPS)*, 2007.

[53] C. Ho, S. Jabbari, and J. W. Vaughan. Adaptive task assignment for crowdsourced classification. In *International Conference on Machine Learning (ICML)*, 2013.

[54] Jeff Howe. The rise of crowdsourcing. *Wired*, 2006.

[55] Kejia Hu, Jason Acimovic, Francisco Erize, Douglas J Thomas, and Jan A Van Mieghem. Forecasting product life cycle curves: Practical approach and empirical analysis. Manufacturing & Service Operations Management, 2017.

[56] Tao Huang, Robert Fildes, and Didier Soopramanien. The value of competitive information in forecasting fmcg retail product sales and the variable selection problem. *European Journal of Operational Research*, 237(2):738–748, 2014.

[57] Tao Huang, Robert Fildes, and Didier Soopramanien. Forecasting retailer product sales in the presence of structural change. European Journal of Operational Research, 2019.

[58] Rajkamal Iyer, Asim Ijaz Khwaja, Erzo FP Luttmer, and Kelly Shue. Screening peers softly: Inferring the quality of small borrowers. *Management Science*, 62(6):1554–1577, 2015.

[59] Bharat A Jain and Omesh Kini. The post-issue operating performance of ipo firms. *The Journal of Finance*, 49(5):1699–1726, 1994.

[60] Kevin G. Jamieson and Robert Nowak. Active ranking using pairwise comparisons. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.

[61] E. Kamar, S. Hacker, and E. Horvitz. Combing human and machine intelligence in large-scale crowdsourcing. In *International Conference on Autonomous Agents and Multiagent System*, 2012.

[62] Yi-hao Kao, Benjamin V Roy, and Xiang Yan. Directed regression. In *Advances in Neural Information Processing Systems*, pages 889–897, 2009.

[63] D. Karger, S. Oh, and D. Shah. Budget-optimal task allocation for reliable crowdsourcing systems. *Operations Research*, 62(1):1–24, 2013.

[64] David R Karger, Sewoong Oh, and Devavrat Shah. Efficient crowdsourcing for multi-class labeling. *ACM SIGMETRICS Performance Evaluation Review*, 41(1):81–92, 2013.

[65] M. Kendall. A new measure of rank correlation. *Biometrika*, 30:81–89, 1938.

[66] Vinod Kumar, S Natarajan, S Keerthana, KM Chinmayi, and N Lakshmi. Credit risk analysis in peer-to-peer lending system. In *Knowledge Engineering and Applications (ICKEA), IEEE International Conference on*, pages 193–196. IEEE, 2016.

[67] Jen-Wei Kuo, Pu-Jen Cheng, and Hsin-Min Wang. Learning to rank from Bayesian decision inference. In *ACM Conference on Information and Knowledge Management*, 2009.

[68] Beatrice Laurent and Pascal Massart. Adaptive estimation of a quadratic functional by model selection. *Annals of Statistics*, pages 1302–1338, 2000.

[69] Hannah Levitt. Personal loans surge to a record high, 2018.

[70] P. Li, C.J.C. Burges, and Q. Wu. Learning to rank using classification and gradient boosting. In *Advances in Neural Information Processing Systems (NIPS)*, 2008.

[71] Mingfeng Lin, Henry C Lucas Jr, and Galit Shmueli. Research commentarytoo big to fail: large samples and the p-value problem. *Information Systems Research*, 24(4):906–917, 2013.

[72] Mingfeng Lin, Nagpurnanand R Prabhala, and Siva Viswanathan. Judging borrowers by the company they keep: Friendship networks and information asymmetry in online peer-to-peer lending. *Management Science*, 59(1):17–35, 2013.

[73] Q. Liu, J. Peng, and A. Ihler. Variational inference for crowdsourcing. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[74] T.Y. Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3:225–331, 2009.

[75] Tyler Lu and Craig Boutilier. Effective sampling and learning for mallows models with pairwise-preference data. *Journal of Machine Learning Research*, 15(1):3783–3829, 2014.

[76] R.D. Luce. *Individual choice behavior: a theoretical analysis.* Wiley, 1959.

[77] Sandrine Macé and Scott A Neslin. The determinants of pre-and postpromotion dips in sales of frequently purchased goods. *Journal of Marketing Research*, 41(3):339–350, 2004.

[78] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1(14), pages 281–297. Oakland, CA, USA, 1967.

[79] Milad Malekipirbazari and Vural Aksakalli. Risk assessment in social lending via random forests. *Expert Systems with Applications*, 42(10):4621–4631, 2015.

[80] C. L. Mallows. Non-null ranking models. *Biometrika*, 44:114–130, 1957.

[81] Jose L Marroquin and Federico Girosi. Some extensions of the k-means algorithm for image segmentation and pattern classification. Technical report, MASSACHUSETTS INST OF TECH CAMBRIDGE ARTIFICIAL INTELLIGENCE LAB, 1993.

[82] Peter McCullagh. *Generalized linear models.* Routledge, 2019.

[83] Javier Mencía. Assessing the risk-return trade-off in loan portfolios. *Journal of Banking & Finance*, 36(6):1665–1677, 2012.

[84] Sounaka Mishra and Kripasindhu Sikdar. On approximability of linear ordering and related np-optimizationproblems ongraphs. *Discrete Applied Mathematics*, 136(2–3):249–269, 2004.

[85] Sahand Negahban, Sewoong Oh, and Devavrat Shah. Rank centrality: ranking from pair-wise comparisons. arXiv:1209.1688, 2012.

[86] J. Paisley, D. Blei, and M. Jordan. Variational bayesian inference with stochastic search. In *International Conference on Machine Learning (ICML)*, 2012.

[87] L'uboš Pástor, Lucian A Taylor, and Pietro Veronesi. Entrepreneurial learning, the ipo decision, and the post-ipo drop in firm profitability. *The Review of Financial Studies*, 22(8):3005–3046, 2008.

[88] Thomas Pfeiffer, Xi Alice Gao, Yiling Chen, Andrew Mao, and David G. Rand. Adaptive polling for information aggregation. In *AAAI Conference on Artificial Intelligence*, 2012.

[89] Warron B. Powell. *The Knowledge Gradient for Optimal Learning.* Wiley Encyclopedia for Operations Research and Management Science, 2010.

[90] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* Wiley, 2005.

[91] Li Qian, Jinyang Gao, and H. V. Jagadish. Learning user preferences by adaptive pairwise comparison. *Proceedings of the VLDB Endowment*, 8(11):1322–1333, 2015.

[92] T. Qin, X. Geng, and T. Y. Liu. A new probabilistic model for rank aggregation. In *Advances in Neural Information Processing Systems (NIPS)*, 2010.

[93] Kira Radinsky and Nir Ailon. Ranking from pairs and triplets: Information quality, evaluation methods and query complexity. In *ACM International Conference on Web Search and Data Mining*, 2011.

[94] Arun Rajkumar and Shivani Agarwal. A statistical convergence perspective of algorithms for rank aggregation from pairwise data. In *International Conference on Machine Learning (ICML)*, 2014.

[95] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy. Learning from crowds. *Journal of Machine Learning Research*, 11(4):1297–1322, 2010.

[96] Oren Rigbi. The effects of usury laws: Evidence from the online loan market. *Review of Economics and Statistics*, 95(4):1238–1248, 2013.

[97] Ilya O. Ryzhov, Warren B. Powell, and Peter I. Frazier. The knowledge gradient algorithm for a general class of online learning problems. *Operations Research*, 60(1):180–195, 2012.

[98] Nihar B. Shah, Sivaraman Balakrishnan, Joseph Bradley, Abhay Parekh, Kannan Ramchandran, and Martin J. Wainwright. Estimation from pairwise comparisons: Sharp minimax bounds with topology dependence. *Journal of Machine Learning Research*, 17, 2016.

[99] Nihar B. Shah, Sivaraman Balakrishnan, Adityanand Guntuboyina, and Martin J. Wainwright. Stochastically transitive models for pairwise comparisons: Statistical and computational issues. In *International Conference on Machine Learning (ICML)*, 2016.

[100] Justin Sirignano and Kay Giesecke. Risk analysis for large pools of loans. *Management Science*, 2018.

[101] Justin A Sirignano, Gerry Tsoukalas, and Kay Giesecke. Large-scale loan portfolio selection. *Operations Research*, 64(6):1239–1255, 2016.

[102] Elizabeth A Stuart, Haiden A Huskamp, Kenneth Duckworth, Jeffrey Simmons, Zirui Song, Michael E Chernew, and Colleen L Barry. Using propensity scores in difference-in-differences models to estimate the effects of a policy change. *Health Services and Outcomes Research Methodology*, 14(4):166–182, 2014.

[103] Michael Taylor, John Guiver, Stephen Robertson, and Tom Minka. Softrank: Optimising non-smooth rank metrics. In *ACM International Conference on Web Search and Data Mining (WSDM)*, 2008.

[104] Siew Hong Teoh, Tak J Wong, and Gita R Rao. Are accruals during initial public offerings opportunistic? *Review of accounting studies*, 3(1-2):175–208, 1998.

[105] L. L. Thurstone. The method of paired comparisons for social values. *Journal of Abnormal and Social Psychology*, 21:384–400, 1927.

[106] Andrew Vakhutinsky, Kresimir Mihic, and Su-Ming Wu. A prescriptive analytics approach to markdown pricing for an e-commerce retailer. Working paper, 2018.

[107] Harald J Van Heerde, Peter SH Leeflang, and Dick R Wittink. The estimation of pre-and postpromotion dips with store-level scanner data. *Journal of Marketing Research*, 37(3):383–395, 2000.

[108] Maksims N. Volkovs and Richard S. Zemel. New learning methods for supervised and unsupervised preference aggregation. *Journal of Machine Learning Research*, 15(1):1135–1176, 2014.

[109] Haizhou Wang and Mingzhou Song. Ckmeans. 1d. dp: optimal k-means clustering in one dimension by dynamic programming. *The R journal*, 3(2):29, 2011.

[110] Fabian Wauthier, Michael Jordan, and Nebojsa Jojic. Efficient ranking from pairwise comparisons. In *International Conference on Machine Learning (ICML)*, 2013.

[111] Zaiyan Wei and Mingfeng Lin. Market mechanisms in online peer-to-peer lending. *Management Science*, 63(12):4236–4257, 2016.

[112] P. Welinder, S. Branson, S. Belongie, and P. Perona. The multidimensional wisdom of crowds. In *Advances in Neural Information Processing Systems (NIPS)*, 2010.

[113] J. Whitehill, P. Ruvolo, T. Wu, J. Bergsma, and J. R. Movellan. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in Neural Information Processing Systems (NIPS)*, 2009.

[114] Coady Wing, Kosali Simon, and Ricardo A Bello-Gomez. Designing difference in difference studies: Best practices for public health policy research. *Annual Review of Public Health*, 39:453–469, 2018.

[115] Jian Wu and Peter I Frazier. The parallel knowledge gradient method for batch bayesian optimization. arXiv:1606.04414, 2016.

[116] Jun Xu and Hang Li. AdaRank: A boosting algorithm for information retrieval. In *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2007.

[117] Jinfeng Yi, Rong Jin, Shaili Jain, and Anil K. Jain. Inferring users' preferences from crowdsourced pairwise comparisons: A matrix completion approach. In *Conference on Human Computation and Crowdsourcing (HCOMP)*, 2013.

[118] Juanjuan Zhang and Peng Liu. Rational herding in microloan markets. *Management Science*, 58(5):892–912, 2012.

[119] Yuchen Zhang, Xi Chen, Dengyong Zhou, and Michael I. Jordan. Spectral methods meet em: A provably optimal algorithm for crowdsourcing. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[120] Zhaohui Zheng, Hongyuan Zha, Tong Zhang, Olivier Chapelle, Keke Chen, and Gordon Sun. A general boosting method and its application to learning ranking functions for web search. In *Advances in Neural Information Processing Systems (NIPS)*. 2008.