Data-Driven Optimization in Revenue Management: Pricing, Assortment Planning, and Demand Learning

by

Sentao Miao

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Industrial and Operations Engineering)
in the University of Michigan
2020

Doctoral Committee:

Professor Xiuli Chao, Chair
Professor Roman Kapuscinski
Professor Viswanath Nagarajan
Professor Cong Shi

Sentao Miao
semiao@umich.edu
ORCID iD: 0000-0002-0380-0797

This dissertation is dedicated to all the people around the globe fighting COVID-19.

Acknowledgments

First of all, I want to thank my advisor and dissertation chair Prof. Xiuli Chao for his time and efforts of guiding me through my six years as a PhD student. For me, he is not only an academic advisor, but also a role model and a mentor of life. I also want to thank Prof. Xi Chen from New York University, who contributed to one chapter of this thesis and generously funded my research for two summers. My appreciation goes to Jiaxi Liu and Yidong Zhang from Alibaba as well, who supported my research by providing with real data and assisting in a field experiment at Alibaba. I would also like to recognize the invaluable discussion and feedback provided by my committee members Prof. Roman Kapuscinski, Prof. Viswanath Nagarajan, and Prof. Cong Shi.

Second, my deepest gratitude goes to my dearest parents, especially my mother. Their selfless support is the key for everything I have achieved in my life. My mother, in particular, is a strong woman who supported the family throughout the difficult time and gave me the best education of life by everything she did.

Moreover, I would like to thank the faculty I met at the University of Michigan especially Prof. Stefanus Jasin and Prof. Jacob Abernathy from whom I learned a lot through research and course. I also appreciate all the feedbacks and suggestions from the friends I met in and outside Ann Arbor, in particular Huanan Zhang, Boxiao (Beryl) Chen, Jingchen Wu, Xiting Gong, and Sean X. Zhou, who are all former students of my advisor.

Last but not least, I would like to pay my special tribute to Kobe Bryant, who tragically died at a young age in a helicopter crash in January 2020. He was a great athlete and entrepreneur, and has been an inspiration since my teenage. His Mamba mentality encourages me to keep chasing my dream no matter what the difficulty is. May he, his daughter Gigi, and other passengers died in that tragedy rest in peace.

TABLE OF CONTENTS

List of Figures

Figure

List of Tables

Table

Abstract

This dissertation studies several problems in revenue management involving dynamic pricing, assortment selection, and their joint optimization, through demand learning. The setting in these problems is that customers' responses to selling prices and product displays are unknown *a priori*, and the only information the decision maker can observe is sales data. Data-driven optimizing-while-learning algorithms are developed in this thesis for these problems, and the theoretical performances of the algorithms are established. For each algorithm, it is shown that as sales data accumulate, the average revenue achieved by the algorithm converges to the optimal.

Chapter 2 studies the problem of context-based dynamic pricing of online products, which have low sales. For these products, existing single-product dynamic pricing algorithms do not work well due to insufficient data samples. To address this challenge, we propose pricing policies that concurrently perform clustering over products and set individual pricing decisions on the fly. By clustering data and identifying products that have similar demand patterns, we utilize sales data from products within the same cluster to improve demand estimation for better pricing decisions. We evaluate the algorithms using regret, and the result shows that when product demand functions come from multiple clusters, our algorithms significantly outperform traditional single-product pricing policies. Simulations with both synthetic and real data from Alibaba show that our algorithm performs very well, and a field experiment at Alibaba shows that our algorithm increased the overall revenue by 10.14%.

Chapter 3 investigates an online personalized assortment optimization problem where

customers arrive sequentially and make their choices (e.g., click an ad, purchase a product) following the multinomial logit (MNL) model with unknown parameters. We develop several algorithms to tackle this problem where the number of data samples is huge and customers' data are possibly high dimensional. Theoretical performance for our algorithms in terms of regret are derived, and numerical experiments on a real dataset from Yahoo! on news article recommendation show that our algorithms perform very well compared with benchmarks.

Chapter 4 considers a joint assortment optimization and pricing problem where customers arrive sequentially and make purchasing decisions following the multinomial logit (MNL) choice model. Not knowing the customer choice parameters *a priori* and subjecting to a display capacity constraint, we dynamically determine the subset of products for display and the selling prices to maximize the expected total revenue over a selling horizon. We design a learning algorithm that balances the trade-off between demand learning and revenue extraction, and evaluate the performance of the algorithm using Bayesian regret. This algorithm uses the method of random sampling to simultaneously learn the demand and maximize the revenue on the fly. An instance-independent upper bound for the Bayesian regret of the algorithm is obtained and numerical results show that it performs very well.

Chapter 1

Introduction

Revenue management has been a popular research area for decades (see e.g., Chen and Chen 2015 for a recent survey) with many success stories in different industries such as retail, airlines, and hotels (see, e.g., Smith et al. 1992, Cross 1995). There are two major problems in the area of revenue management: pricing and assortment optimization. In the problem of pricing, the firm aims to maximize the revenue/profit according to the current understanding of supply and demand with respect to different prices. For assortment optimization, the firm selects a subset from a whole universe of products to put on its shelf/webpage for revenue/profit maximization. In most of the existing literature, both problems are studied in the setting that demand is known to the firm until recently (den Boer 2015, Kök et al. 2015). However, this known demand information assumption is usually not appropriate in today's rapidly changing market. This thesis specifically addresses the problems of dynamic pricing, assortment optimization, as well as their joint optimization when demand is not known *a priori*.

In Chapter 2, we consider the problem of dynamic pricing for products with low sales or popularity through demand learning. Data from Alibaba, the largest global online retailer, shows that most of the products on its website belongs to the category of unpopular products, i.e., very few customers view those products each day. Pricing low-sale products is often challenging due to the limited sales records available for demand estimation. In this chapter, we tackle this challenge using the method of clustering. More specifically, although each low-sale product only has a few sales records, the total number of low-sale products is usually quite large. Our starting point is that there are some set of products out there, though we do not know which ones, that share similar underlying demand patterns. For these products, information can be extracted from their collective sales data to improve the estimation of their demand function. Using clustering, we develop adaptive learning algorithms that identify the products exhibiting similar demand patterns and learn their demand jointly for revenue maximization. Theoretical performances of our algorithms are proved to be promising, and numerical experiments using both synthetic and real data from

Alibaba also show that our algorithms outperform several benchmarks in the existing literature. In the end, we implemented our algorithm in a field experiment at Alibaba, and it achieved a revenue increase of 10.14%.

Chapter 3 investigates a personalized assortment optimization problem. More specifically, the firm needs to select a subset of items (e.g., products, ads) and offer to each customer tailored to his/her personal preference. Using customer's personal data including gender, age, location, browsing history, the firm can extract useful information in order to match with the most preferred items. However, in this problem the decision maker usually faces the challenge of *big data*. That is, the number of customers is very large, making the historical data accumulate as customers keep arriving. Moreover, the customer's data is usually extremely high dimensional, hence making the recommendation decision for each customer time consuming. This chapter addresses this challenge of big data by developing adaptive algorithms which combine the method of online convex optimization for demand learning and dimension reduction through random projection. Both theoretical and empirical performances of our algorithms are developed, and they are shown to be promising.

In Chapter 4, we study the joint pricing and assortment optimization with demand learning. As we have illustrated, both pricing and assortment optimization are important research topics in revenue management with abundant literature. However, the research on their joint optimization is surprisingly scarce. In this chapter, we study the dynamic joint assortment optimization and pricing problem when customers follow the multinomial logit (MNL) choice model. An algorithm based on Thompson sampling is developed which balances the trade-off between demand exploration and revenue exploitation. We evaluate the performance of the algorithm using the so-called Bayesian regret, and results show that its Bayesian regret upper bound is very close to the regret lower bound. Several numerical experiments based on synthetic data are also conducted, and our algorithm significantly outperforms the benchmarks.

The thesis concludes in Chapter 5 with a summary of the key findings and an outline of some opportunities for future research that stem from this work.

Chapter 2

Context-Based Dynamic Pricing with Online Clustering

2.1 Introduction

Over the past several decades, dynamic pricing has been widely adopted by industries, such as retail, airlines, and hotels, with great success (see, e.g., Smith et al. 1992, Cross 1995). Dynamic pricing has been recognized as an important lever not only for balancing supply and demand, but also for increasing revenue and profit. Recent advances in online retailing and increased availability of online sales data have created opportunities for firms to better use customer information to make pricing decisions, see e.g., the survey paper by den Boer (2015). Indeed, the advances in information technology have made the sales data easily accessible, facilitating the estimation of demand and the adjustment of price in real time. Increasing availability of demand data allows for more knowledge to be gained about the market and customers, as well as the use of advanced analytics tools to make better pricing decisions.

However, in practice, there are often products with low sales amount or user views. For these products, few available data points exist. For example, *Tmall Supermarket*, a business division of Alibaba, is a large-scale online store. In contrast to a typical consumer-to-consumer (C2C) platform (e.g., Taobao under Alibaba) that has millions of products available, Tmall Supermarket is designed to provide carefully selected high-quality products to customers. We reviewed the sales data from May to July of 2018 on Tmall Supermarket with nearly 75,000 products offered during this period of time, and it shows that more than 16,000 products (21.6% of all products) have a daily average number of unique visitors[1] less than 10, and more than 10,000 products (14.3% of all products) have a daily average number of unique visitors less than or equal to 2. Although each low-sale product

[1] A terminology used within Alibaba to represent a unique user login identification.

alone may have little impact on the company's revenue, the combined sales of all low-sale products are significant.

Pricing low-sale products is often challenging due to the limited sales records available for demand estimation. In fast-evolving markets (e.g., fashion or online advertising), demand data from the distant past may not be useful for predicting customers' purchasing behavior in the near future. Classical statistical estimation theory has shown that data insufficiency leads to large estimation error of the underlying demand, which results in sub-optimal pricing decisions. In fact, the research on dynamic pricing of products with little sales data remains relatively unexplored. To the best of our knowledge, there exists no dynamic pricing policy in the literature for low-sale products that admits theoretical performance guarantee. Our research fills the gap by developing adaptive context-based dynamic pricing learning algorithms for low-sale products, and our results show that the algorithms perform well both theoretically and numerically (including a field experiment).

2.1.1 Contributions of the chapter

Although each low-sale product only has a few sales records, the total number of low-sale products is usually quite large. In this chapter, we address the challenge of pricing low-sale products using an important idea from machine learning — clustering. Our starting point is that there are some set of products out there, though we do not know which ones, that share similar underlying demand patterns. For these products, information can be extracted from their collective sales data to improve the estimation of their demand function. The problem is formulated as developing adaptive learning algorithms that identify the products exhibiting similar demand patterns, and extract the hidden information from sales data of seemingly unrelated products to improve the pricing decisions of low-sale products and increase revenue.

We first consider a generalized linear demand model with stochastic contextual covariate information about products and develop a learning algorithm that integrates product clustering with pricing decisions. Our policy consists of two phases. The first phase constructs confidence bounds on the distance between clusters, which enables dynamic clustering without any prior knowledge of the cluster structure. The second phase carefully controls the price variation based on the estimated clusters, striking a proper balance between price exploration and revenue maximization by exploiting the cluster structure. Since the pricing part of the algorithm is inspired by semi-myopic policy proposed by Keskin and Zeevi (2014), we refer to our algorithm as the *Clustered Semi-Myopic Pricing* (CSMP) policy. We first establish the theoretical regret bound of the proposed policy.

Specifically, when the demand functions of the products belong to m clusters, where m is smaller than the total number of products (denoted by n), the performance of our algorithm is better than that of existing dynamic pricing policies that treat each product separately. Let T denote the length of the selling season; we show in Theorem 2.3.1 that our algorithm achieves the regret of $\tilde{O}(\sqrt{mT})$, where $\tilde{O}(\cdot)$ hides the logarithmic terms. This result, when m is much smaller than n, is a significant improvement over the regret when applying a single-product pricing policy to individual products, which is typically $\tilde{O}(\sqrt{nT})$.

When the demand function is linear in terms of covariates of products and price, we extend our result to the setting where the covariates are non-stochastic and even adversarial. In this case, we develop a variant of the CSMP policy (called CSMP-L, where L stands for "linear"), which handles a more general class of demand covariates. The parameter estimation for the linear demand function is based on a scheme developed by Nambiar et al. (2018), which is used to build separate confidence bounds for the parameters of demand covariates and price sensitivity. Similar to the CSMP algorithm, our theoretical analysis in Theorem 2.4.1 shows that the CSMP-L algorithm achieves the regret $\tilde{O}(\sqrt{mT})$.

We carry out a thorough numerical experiment using both synthetic data and a real dataset from Alibaba consisting of a large number of low-sale products. Several benchmarks, one treats each product separately, one puts all products into a single cluster, and the other one applies a classical clustering method (K-means method for illustration), are compared with our algorithms under various scenarios. The numerical results show that our algorithms are effective and their performances are consistent in different scenarios (e.g., with almost static covariates, model misspecification).

Our algorithm was tested in a field experiment conducted at Alibaba by a Tmall Supermarket team. The algorithm was tested on 40 products for 30 consecutive days. The results from the field experiment show that the overall revenue was boosted by 10.14%.

It is well-known that providing a performance guarantee for a clustering method is challenging due to the non-convexity of the loss function (e.g., in K-means), which is why there exists no clustering and pricing policy with theoretical guarantees in the existing literature. This is the first work to establish the regret bound for a dynamic clustering and pricing policy. Instead of adopting an existing clustering algorithm from the machine learning literature (e.g., K-means), which usually requires the number of clusters as an input, our algorithms dynamically update the clusters based on the gathered information about customers' purchase behavior. In addition to significantly improving the theoretical performance as compared to classical dynamic pricing algorithms without clustering, our algorithms demonstrate excellent performance both in our simulation study and in our field experiments with Alibaba.

2.1.2 Literature review

In this subsection, we review some related research from both the revenue management and machine learning literature.

Related literature in dynamic pricing. Due to increasing popularity of online retailing, dynamic pricing has become an active research area in revenue management in the past decade. We only briefly review a few of the most related works and refer the interested readers to den Boer (2015) for a comprehensive literature survey. Earlier work and review of dynamic pricing include Gallego and Van Ryzin (1994, 1997), Bitran and Caldentey (2003), Elmaghraby and Keskinocak (2003). These papers assume that demand information is known to the retailer *a priori* and either characterize or compute the optimal pricing decisions. In some retailing industries, such as fast fashion, this assumption may not hold due to the quickly changing market environment. As a result, with the recent development of information technology, combining dynamic pricing with demand learning has attracted much interest in research. Depending on the structure of the underlying demand functions, these works can be roughly divided into two categories: parametric demand models (see, e.g., Carvalho and Puterman 2005, Bertsimas and Perakis 2006, Besbes and Zeevi 2009, Farias and Van Roy 2010, Broder and Rusmevichientong 2012, Harrison et al. 2012, den Boer and Zwart 2013, Keskin and Zeevi 2014) and nonparametric demand models (see, e.g., Araman and Caldentey 2009, Wang et al. 2014, Lei et al. 2014, Chen et al. 2015a, Besbes and Zeevi 2015, Cheung et al. 2017, Chen and Shi 2019). The aforementioned papers assume that the price is continuous. Other works consider a discrete set of prices, see, e.g., Ferreira et al. (2018a), and recent studies examine pricing problems in dynamically changing environments, see, e.g., Besbes et al. (2015) and Keskin and Zeevi (2016).

Dynamic pricing and learning with demand covariates (or contextual information) has received increasing attention in recent years because of its flexibility and clarity in modeling customers and market environment. Research involving this information include, among others, Chen et al. (2015b), Qiang and Bayati (2016), Nambiar et al. (2018), Ban and Keskin (2017), Lobel et al. (2018), Chen and Gallego (2018), Javanmard and Nazerzadeh (2019). In many online-retailing applications, sellers have access to rich covariate information reflecting the current market situation. Moreover, the covariate information is not static but usually evolves over time. This work incorporates time-evolving covariate information into the demand model. In particular, given the observable covariate information of a product, we assume that the customer decision depends on both the selling price and covariates. Although covariates provide richer information for accurate demand estimation, a demand model that incorporates covariate information involves more parameters to be estimated. Therefore, it requires more data for estimation with the presence of

covariates, which poses an additional challenge for low-sale products.

Related literature in clustering for pricing. To the best of our knowledge, we are not aware of any operations literature that dynamically learns about the clustering structure on the fly. There are, however, some interesting works that use historical data to determine the cluster structure of demand functions in an offline manner, and then dynamically make pricing decisions for another product by learning which cluster its demand belongs to.

Ferreira et al. (2015) study a pricing problem with flash sales on the Rue La La platform. Using historical information and offline optimization, the authors classify the demand of all products into multiple groups, and use demand information for products that did not experience lost sales to estimate demand for products that had lost sales. They construct "demand curves" on the percentage of total sales with respect to the number of hours after the sales event starts, then classify these curves into four clusters. For a sold-out product, they check which one of the four curves is the closest to its sales behavior and use that to estimate the lost sales. Cheung et al. (2017) consider the single-product pricing problem, where the demand of the product is assumed to be from one of the K demand functions (called *demand hypothesis* in that paper). Those K demand functions are assumed to be known, and the decision is to choose which of those functions is the true demand curve of the product. In their field experiment with Groupon, they applied K-means clustering to historical demand data to generate those K demand functions offline. That is, clustering is conducted offline first using historical data, then dynamic pricing decisions are made in an online fashion for a new product, assuming that its demand is one of the K demand functions.

Related literature in other operations management problems. The method of clustering is quite popular for many operations management problems such as demand forecast for new products and customer segmentation. In the following, we give a brief review of some recent papers on these two topics that are based on data clustering approach.

Demand forecasting for new products is a prevalent yet challenging problem. Since new products at launch have no historical sales data, a commonly used approach is to borrow data from "similar old products" for demand forecasting. To connect the new product with old products, current literature typically use product features. For instance, Baardman et al. (2017) assume a demand function which is a weighted sum of unknown functions (each representing a cluster) of product features. While in Ban et al. (2018), similar products are predefined such that common demand parameters are estimated using sales data of old products. Hu et al. (2018) investigate the effectiveness of clustering based on product category, features, or time series of demand respectively.

Customer segmentation is another application of clustering. Jagabathula et al. (2018)

assume a general parametric model for customers' features with unknown parameters, and use K-means clustering to segment customers. Bernstein et al. (2018) consider the dynamic personalized assortment optimization using clustering of customers. They develop a hierarchical Bayesian model for mapping from customer profiles to segments.

Compared with these literature, besides a totally different problem setting, this chapter is also different in the approach. First, we consider an online clustering approach with provable performance instead of an offline setting as in Baardman et al. (2017), Ban et al. (2018), Hu et al. (2018), Jagabathula et al. (2018). Second, we know neither the number of clusters (in contrast to Baardman et al. 2017, Bernstein et al. 2018 that assume known number of clusters), nor the set of products in each cluster (as compared with Ban et al. 2018 who assume known products in each cluster). Finally, we do not assume any specific probabilistic structure on the demand model and clusters (in contrast with Bernstein et al. 2018 who assign and update the probability for a product to belong to some cluster), but define clusters using product neighborhood based on their estimated demand parameters.

Related literature in multi-arm bandit problem. A successful dynamic pricing algorithm requires a careful balancing between exploration (i.e., learning the underlying demand function) and exploitation (i.e., making the optimal pricing strategy based on the learned information so far). The exploration-exploitation trade-off has been extensively investigated in the multi-armed bandit (MAB) literature; see earlier works by Lai and Robbins (1985), Auer et al. (2002), Auer (2002) and Bubeck et al. (2012) for a comprehensive literature review. Among the vast MAB literature, there is a line of research on bandit clustering that addresses a different but related problem (see, e.g., Cesa-Bianchi et al. 2013, Gentile et al. 2014, Nguyen and Lauw 2014, Gentile et al. 2016). The setting is that there is a finite number of arms which belong to several unknown clusters, where unknown reward functions of arms in each cluster are the same. Under this assumption, the MAB algorithms aim to cluster different arms and learn the reward function for each cluster.The setting of the bandit-clustering problem is quite different from ours. In the bandit clustering problem, the arms belong to different clusters and the decision for each period is which arm to play. In our setting, the products belong to different clusters and the decision for each period is what prices to charge for all products, and we have a *continuum set* of prices to choose from for each product. In addition, in contrast to the linear reward in bandit-clustering problem, the demand functions in our setting follow a generalized linear model. As will be seen in Section 2.3, we design a price perturbation strategy based on the estimated cluster, which is very different from the algorithms in bandit-clustering literature.

Related literature in clustering. We end this section by giving a brief overview of clustering methods in the machine learning literature. To save space, we only discuss sev-

eral popular clustering methods, and refer the interested reader to Saxena et al. (2017) for a recent literature review on the topic. The first one is called hierarchical clustering (Murtagh 1983), which iteratively clusters objects (either bottom-up, from a single object to several big clusters; or top-down, from a big cluster to single product). Comparable with hierarchical clustering, another class of clustering method is partitional clustering, in which the objects do not have any hierarchical structure, but rather are grouped into different clusters horizontally. Among these clustering methods, K-means clustering is probably the most well-known and most widely applied method (see e.g., MacQueen et al. 1967, Hartigan and Wong 1979). Several extensions and modifications of K-means clustering method have been proposed in the literature, e.g., K-means++ (Arthur and Vassilvitskii 2007, Bahmani et al. 2012) and fuzzy c-means clustering (Dunn 1973, Bezdek 2013). Another important class of clustering method is based on graph theory. For instance, the spectral clustering uses graph Laplacian to help determine clusters (Shi and Malik 2000, Von Luxburg 2007). Beside these general methods for clustering, there are many clustering methods for specific problems such as decision tree, neural network, etc. It should be noted that nearly all the clustering methods in the literature are based on offline data. This chapter, however, integrates clustering into online learning and decision-making process.

2.1.3 Organization of the chapter

The remainder of this chapter is organized as follows. In Section 2.2, we present the problem formulation. Our main algorithm is presented in Section 2.3 together with the theoretical results for the algorithm performance. We develop another algorithm for the linear demand model in Section 2.4 when the contextual covariates are non-stochastic or adversarial. In Section 2.5, we report the results of several numerical experiments based on both synthetic data and a real dataset in addition to the findings from a field experiment carried out at Alibaba's Tmall Supermarket. We conclude the chapter with a discussion about future research in Section 2.7. Finally, all the technical proofs are presented in the supplement.

2.2 Problem Formulation

We consider a retailer that sells n products, labeled by $i = 1, 2, \ldots, n$, with unlimited inventory (e.g., there is an inventory replenishment scheme such that products typically do not run out of stock). Following the literature, we denote the set of these products by $[n]$. We mainly focus on online retailing of low-sale products. These products are typically not of-

fered to customers as a display; hence we do not consider substitutability/complementarity of products in our model. Furthermore, these products are usually not recommended by the retailer on the platform, and instead, customers search for them online. We let $q_i > 0$ denote the percentage of potential customers who are interested in, or search for, product $i \in [n]$. In this chapter, we will treat q_i as the probability an arriving customer searches for product i.

Customers arrive sequentially at time $t = 1, 2, \ldots, T$, and we denote the set of all time indices by $[T]$. For simplicity, we assume without loss of generality that there is exactly one arrival during each period. In each time period t, the firm first observes some covariates for each product i, such as product rating, prices of competitors, average sales in past few weeks, and promotion-related information (e.g., whether the product is currently on sale). We denote the covariates of product i by $z_{i,t} \in \mathbb{R}^d$, where d is the dimension of the covariates that is usually small (as compared to n or T). The covariates $z_{i,t}$ change over time and satisfy $||z_{i,t}||_2 \leq 1$ after normalization. Then, the retailer sets the price $p_{i,t} \in [\underline{p}, \overline{p}]$ for each product i, where $0 \leq \underline{p} < \overline{p} < \infty$ (the assumption of the same price range for all products is without loss of generality). Let i_t denote the product that the customer searches in period t (or customer t). After observing the price and other details of product i_t, customer t then decides whether or not to purchase it. The sequence of events in period t is summarized as follows:

i) In time t, the retailer observes the covariates $z_{i,t}$ for each product $i \in [n]$, then sets the price $p_{i,t}$ for each $i \in [n]$.

ii) Customer searches for product $i_t \in [n]$ in period t with probability q_{i_t} independent of others and then observes its price $p_{i_t,t}$.

iii) The customer decides whether or not to purchase product i_t.

The customer's purchasing decision follows a *generalized linear model* (GLM, see e.g., McCullagh and Nelder 1989). That is, given price $p_{i_t,t}$ of product i_t at time t, the customer's purchase decision is represented by a Bernoulli random variable $d_{i_t,t}(p_{i_t,t}; z_{i_t,t}) \in \{0, 1\}$, where $d_{i_t,t}(p_{i_t,t}; z_{i_t,t}) = 1$ if the customer purchases product i_t and 0 otherwise. The purchase probability, which is the expectation of $d_{i_t,t}(p_{i_t,t}; z_{i_t,t})$, takes the form

$$\mathbb{E}[d_{i_t,t}(p_{i_t,t}; z_{i_t,t})] = \mu(\alpha'_{i_t} x_{i_t,t} + \beta_{i_t} p_{i_t,t}), \tag{2.1}$$

where $\mu(\cdot)$ is the link function, $x'_{i_t,t} = (1, z'_{i_t,t})$ is the corresponding extended demand covariate with the 1 in the first entry used to model the bias term in a GLM model, and the

expectation is taken with respect to customer purchasing decision. Let $\theta'_{i_t} = (\alpha'_{i_t}, \beta_{i_t})$ be the unknown parameter of product i_t, which is assumed to be bounded. That is, $||\theta_i||_2 \leq L$ for some constant L for all $i \in [n]$.

Remark 2.2.1 *The commonly used linear and logistic models are special cases of GLM with link function $\mu(x) = x$ and $\mu(x) = \exp(x)/(1+\exp(x))$, respectively. The parametric demand model* (2.1) *has been used in a number of papers on pricing with contextual information, see, e.g., Qiang and Bayati (2016) (for a special case of linear demand with $\mu(x) = x$) and Ban and Keskin (2017).*

For convenience and with a slight abuse of notation, we write

$$p_t := p_{i_t,t}, \quad z_t := z_{i_t,t}, \quad x_t := x_{i_t,t}, \quad d_t := d_{i_t,t},$$

where " := " stands for "defined as". Let the feasible sets of x_t and θ_i be denoted as $\mathcal{X}$ and Θ, respectively. We further define

$$\mathcal{T}_{i,t} := \{s \leq t \ : i_s = i\} \tag{2.2}$$

as the set of time periods before t in which product i is viewed, and $T_{i,t} := |\mathcal{T}_{i,t}|$ its cardinality. With this demand model, the expected revenue $r_t(p_t)$ of each round t is

$$r_t(p_t) := p_t \mu(\alpha'_{i_t} x_t + \beta_{i_t} p_t). \tag{2.3}$$

Note that we have made the dependency of $r_t(p_t)$ on x_t implicit.

The firm's optimization problem and regret. The firm's goal is to decide the price $p_t \in [\underline{p}, \overline{p}]$ at each time t for each product to maximize the cumulative expected revenue $\sum_{t=1}^{T} \mathbb{E}[r_t(p_t)]$, where the expectation is taken with respect to the randomness of the pricing policy as well as the stream of i_t for $t \in [T]$, and for the next section, also the stochasticity in contextual covariates z_t, $t \in [T]$. The goal of maximizing the expected cumulative revenue is equivalent to minimizing the so-called regret, which is defined as the revenue gap as compared with the *clairvoyant decision maker* who knew the underlying parameters in the demand model *a priori*. With the known demand model, the optimal price can be computed as

$$p_t^* = \underset{p \in [\underline{p}, \overline{p}]}{\arg\max}\, r_t(p),$$

and the corresponding revenue gap at time t is $\mathbb{E}[r_t(p_t^*) - r_t(p_t)]$ (the dependency of p_t^* on x_t is again made implicit). The cumulative regret of a policy π with prices $\{p_t\}_{t=1}^{T}$ is

defined by the summation of revenue gaps over the entire time horizon, i.e.,

$$R^{\pi}(T) := \sum_{t=1}^{T} \mathbb{E}[r_t(p_t^*) - r_t(p_t)]. \tag{2.4}$$

Remark 2.2.2 *For consistency with the online pricing literature, see e.g., Chen et al. (2015b), Qiang and Bayati (2016), Ban and Keskin (2017), Javanmard and Nazerzadeh (2019), in this chapter we use expected revenue as the objective to maximize. However, we point out that all our analyses and results carry over to the objective of profit maximization. That is, if c_t is the cost of the product in round t, then the expected profit in (2.3) can be replaced by*

$$r_t(p_t) = (p_t - c_t)\mu(\alpha'_{i_t} x_t + \beta_{i_t} p_t).$$

Cluster of products. Two products i_1 and i_2 are said to be "similar" if they have similar underlying demand functions, i.e., θ_{i_1} and θ_{i_2} are close. In this chapter we assume that the n products can be partitioned into m clusters, $\mathcal{N}_j$ for $j = 1, 2, \ldots, m$, such that for arbitrary two products i_1 and i_2, we have $\theta_{i_1} = \theta_{i_2}$ if i_1 and i_2 belong to the same cluster; otherwise, $||\theta_{i_1} - \theta_{i_2}||_2 \geq \gamma > 0$ for some constant γ. We refer to this cluster structure as the γ-gap assumption, which will be relaxed in Remark 2.3.4 of Section 2.3.2. For convenience, we denote the set of clusters by $[m]$, and by a bit abuse of notation, let $\mathcal{N}_i$ be the cluster to which product i belongs.

It is important to note that the number of clusters m and each cluster $\mathcal{N}_j$ are *unknown* to the decision maker *a priori*. Indeed, in some applications such structure may not exist at all. If such structure does exist, then our policy can identify such a cluster structure and make use of it to improve the practical performance and the regret bound. However, we point out that the cluster structure is not a requirement for the pricing policy to be discussed. In other words, our policy reduces to a standard dynamic pricing algorithm when demand functions of the products are all different (i.e., when $m = n$).

It is also worthwhile to note that our clustering is based on demand parameters/patterns and *not* on product categories or features, since it is the demand of the products that we want to learn. The clustering approach based on demand is prevalent in the literature (besides Ferreira et al. 2015, Cheung et al. 2017 and the references therein, we also refer to Van Kampen et al. 2012 for a comprehensive review). Clustering based on category/feature similarity is useful in some problems (see e.g., Su and Chen 2015 investigate customer segmentation using features of clicking data), but it does not apply to our setting, because, for instance, products with similar feature for different brands may have very different demand.

Remark 2.2.3 *For its application to the online pricing problem, the contextual information in our model is about the product. That is, at the beginning of each period, the firm observes the contextual information about each product, then determines the pricing decision for the product, and then the arriving customer makes a purchasing decisions. We point out that our algorithm and result apply equally to personalized pricing in which the contextual information is about the customer. That is, a customer arrives (e.g., logging on the website) and reveals his/her contextual information, and then the firm makes a pricing decision based on that information. The objective is to make personalized pricing decisions to maximize total revenue (see e.g., Ban and Keskin 2017).*

2.3 Pricing Policy and Main Results

In this section we discuss the specifics of the learning algorithm, its theoretical performance, and a sketch of its proof. Specifically, we describe the policy procedure and discuss its intuitions in Section 2.3.1 before presenting its regret and outlining the proof in Section 2.3.2.

2.3.1 Description of the pricing policy

Our policy consists of two phases for each period $t \in [T]$: the first phase constructs a *neighborhood* for each product $i \in [n]$, and the second phase determines its selling price. In the first step, our policy uses *individual data* of each product $i \in [n]$ to estimate parameters $\hat{\theta}_{i,t-1}$. This estimation is used only for construction of the neighborhood $\hat{\mathcal{N}}_{i,t}$ for product i. Once the neighborhood is defined, we consider all the products in this neighborhood as in the same cluster and use *clustered data* to estimate the parameter vector $\widetilde{\theta}_{\hat{\mathcal{N}}_{i,t},t-1}$. The latter is used in computing the selling price of product i. We refer to Figure 2.1 for a flowchart of our policy, and present the detailed procedure in Algorithm 1.

In the following, we discuss the parameter estimation of GLM demand functions and the construction of a neighborhood in detail.

Parameter estimation of GLM. As shown in Figure 2.1, the parameter estimation is an important part of our policy construction. We adopt the classical maximum likelihood estimation (MLE) method for parameter estimation (see McCullagh and Nelder 1989). For completeness, we briefly describe the MLE method here. Let $u_t := (x_t', p_t)' \in \mathbb{R}^{d+2}$. The conditional distribution of the demand realization d_t, given u_t, belongs to the exponential

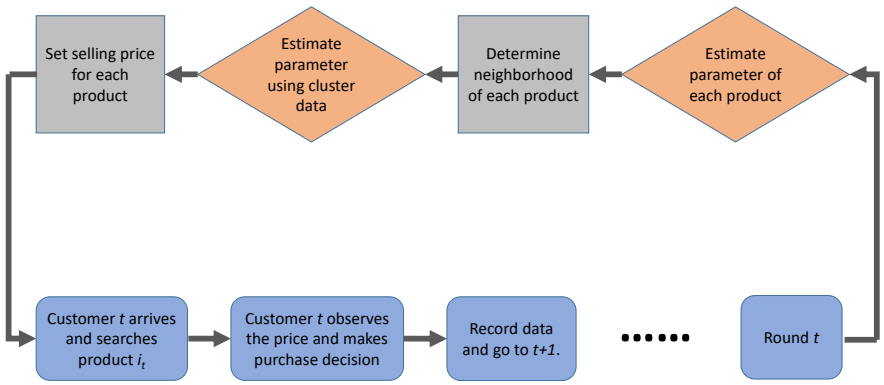


Figure 2.1: Flow chart of the algorithm.

family and can be written as

$$\mathbb{P}(d_t|u_t) = \exp\left(\frac{d_t u_t'\theta - m(u_t'\theta)}{g(\eta)} + h(d_t,\eta)\right). \tag{2.5}$$

Here $m(\cdot)$, $g(\cdot)$, and $h(\cdot)$ are some specific functions, where $\dot{m}(u_t'\theta) = \mathbb{E}[d_t] = \mu(u_t'\theta)$ depends on $\mu(\cdot)$ and $h(d_t,\eta)$ is the normalization part, and η is some known scale parameter. Suppose that we have t samples (d_s, p_s) for $s = 1, 2, \ldots, t$, the negative log-likelihood function of θ under model (2.5) is

$$\sum_{s=1}^{t}\left(\frac{m(u_s'\theta) - d_s u_s'\theta}{g(\eta)} + h(d_s,\eta)\right). \tag{2.6}$$

By extracting the terms in (2.6) that involves θ, the maximum likelihood estimator $\hat{\theta}$ is

$$\hat{\theta} = \arg\min_{\theta\in\Theta}\sum_{s=1}^{t} l_s(\theta), \qquad l_s(\theta) := m(u_s'\theta) - d_s u_s'\theta. \tag{2.7}$$

Since $\nabla^2 l_s(\theta) = \dot{\mu}(u_s'\theta)u_s u_s'$ is positive semi-definite in a standard GLM model (by Assumption A-2 in the next subsection), the optimization problem in (2.7) is convex and can be easily solved.

Determining the neighborhood of each product. The first phase of our policy determines which products to include in the neighborhood of each product $i \in [n]$. We use the term "neighborhood" instead of cluster, though closely related, because clusters are usu-

ally assumed to be disjoint in the machine learning literature. In contrast, by our definition of neighborhood, some products can belong to different neighborhoods depending on the estimated parameters. To define the neighborhood of i, which is denoted by $\hat{\mathcal{N}}_{i,t}$, we first estimate parameter $\hat{\theta}_{i,t-1}$ of each product $i \in [n]$ using their own data, i.e., $\hat{\theta}_{i,t-1}$ is the maximum likelihood estimator using data in $\mathcal{T}_{i,t-1}$ defined in (2.2). Then, we include a product $i' \in [n]$ in the neighborhood $\hat{\mathcal{N}}_{i,t}$ of i if their estimated parameters are *sufficiently close*, which is defined as

$$||\hat{\theta}_{i',t-1} - \hat{\theta}_{i,t-1}||_2 \leq B_{i',t-1} + B_{i,t-1},$$

where $B_{i,t-1}$ is a *confidence bound* for product i given by

$$B_{i,t} := \frac{\sqrt{c(d+2)\log(1+t)}}{\sqrt{\lambda_{\min}(V_{i,t})}}. \tag{2.8}$$

Here, $V_{i,t} := I + \sum_{s \in \mathcal{T}_{i,t}} u_s u_s'$ is the empirical Fisher's information matrix of product $i \in [n]$ at time t and c is some positive constant, which will be specified in our theory development. Note that, by the γ-gap assumption discussed at the end of Section 2.2, the method will work even when $T_{i,t-1}$ only contains a limited number of sales records.

Setting the price of each product. Once we define the (estimated) neighborhood $\hat{\mathcal{N}}_{i,t}$ of $i \in [n]$, we can pool the demand data of all products in $\hat{\mathcal{N}}_{i,t}$ to learn the parameter vector. That is, we let

$$\widetilde{\mathcal{T}}_{\hat{\mathcal{N}}_{i,t},t-1} := \bigcup_{i' \in \hat{\mathcal{N}}_{i,t}} \mathcal{T}_{i',t-1} \quad \text{and} \quad \tilde{T}_{\hat{\mathcal{N}}_{i,t},t-1} := |\widetilde{\mathcal{T}}_{\hat{\mathcal{N}}_{i,t},t-1}|.$$

The clustered parameter vector $\tilde{\theta}_{\hat{\mathcal{N}}_{i,t},t-1}$ is the maximum likelihood estimator using data in $\tilde{\mathcal{T}}_{\hat{\mathcal{N}}_{i,t},t-1}$.

To decide on the price, we first compute $p'_{i,t}$, which is the "optimal price" based on the estimated clustered parameters $\tilde{\theta}_{\hat{\mathcal{N}}_{i,t},t-1}$. Then we restrict $p'_{i,t}$ to the interval $[\underline{p} + |\Delta_{i,t}|, \overline{p} - |\Delta_{i,t}|]$ by the *projection operator*. That is, we compute

$$\tilde{p}_{i,t} = \text{Proj}_{[\underline{p}+|\Delta_{i,t}|, \overline{p}-|\Delta_{i,t}|]}(p'_{i,t}), \quad \text{where} \quad \text{Proj}_{[a,b]}(x) := \min\{\max\{x, a\}, b\}.$$

The reasoning for this restriction is that our final price $p_{i,t}$ will be $p_{i,t} = \tilde{p}_{i,t} + \Delta_{i,t}$, and the projection operator forces the final price $p_{i,t}$ to the range $[\underline{p}, \overline{p}]$. Here, the price perturbation $\Delta_{i,t} = \pm \Delta_0 \tilde{T}_{\hat{\mathcal{N}}_{i,t},t}^{-1/4}$ takes a positive or a negative value with equal probability, where Δ_0 is a positive constant. We add this price perturbation for the purpose of price exploration.

Intuitively, the more price variation we have, the more accurate the parameter estimation will be. However, too much price variation leads to loss of revenue because we deliberately charged a "wrong" price. Therefore, it is crucial to find a balance between these two targets by defining an appropriate $\Delta_{i,t}$.

We note that this pricing scheme belongs to the class of semi-myopic pricing policies defined in Keskin and Zeevi (2014). Since our policy combines clustering with semi-myopic pricing, we refer to it as the *Clustered Semi-Myopic Pricing* (CSMP) algorithm.

Algorithm 1 The CSMP Algorithm

Require: c, the confidence bound parameter; Δ_0, price perturbation parameter;

1: **Step 0. Initialization.** Initialize $\mathcal{T}_{i,0} = \emptyset$ and $V_{i,0} = I$ for all $i \in [n]$. Let $t = 1$ and go to Step 1.

2: **for** $t = 1, 2, \ldots, T$ **do**

3: **Step 1. Individual Parametric Estimation.** Compute the MLE using individual data

$$\hat{\theta}_{i,t-1} = \arg\min_{\theta \in \Theta} \sum_{s \in \mathcal{T}_{i,t-1}} l_s(\theta)$$

for all $i \in [n]$. Go to Step 2.

4: **Step 2. Neighborhood Construction.** Compute the neighborhood of each product i as

$$\hat{\mathcal{N}}_{i,t} = \{i' \in [n] : \ ||\hat{\theta}_{i',t-1} - \hat{\theta}_{i,t-1}||_2 \leq B_{i',t-1} + B_{i,t-1}\}$$

where $B_{i,t-1}$ is defined in (2.8) for each $i \in [n]$. Go to Step 3.

5: **Step 3. Clustered Parametric Estimation.** Compute the MLE using clustered data

$$(\tilde{\alpha}'_{\hat{\mathcal{N}}_{i,t},t-1}, \tilde{\beta}_{\hat{\mathcal{N}}_{i,t},t-1})' = \tilde{\theta}_{\hat{\mathcal{N}}_{i,t},t-1} = \arg\min_{\theta \in \Theta} \sum_{s \in \tilde{\mathcal{T}}_{\hat{\mathcal{N}}_{i,t},t-1}} l_s(\theta)$$

for each $i \in [n]$. Go to Step 4.

6: **Step 4. Pricing.** Compute price for each $i \in [n]$ as

$$p'_{i,t} = \arg\max_{p \in [\underline{p}, \bar{p}]} \ \mu(\alpha'_{\hat{\mathcal{N}}_{i,t},t-1} x_{i,t} + \beta_{\hat{\mathcal{N}}_{i,t},t-1} p)p,$$

then project to $\tilde{p}_{i,t} = \text{Proj}_{[\underline{p}+|\Delta_{i,t}|, \bar{p}-|\Delta_{i,t}|]}(p'_{i,t})$ and offer to the customer price $p_{i,t} = \tilde{p}_{i,t} + \Delta_{i,t}$ where $\Delta_{i,t} = \pm\Delta_0 \tilde{T}^{-1/4}_{\hat{\mathcal{N}}_{i,t},t}$ which takes two signs with equal probability.

7: Then, customer t arrives, searches for product i_t, and makes purchasing decision $d_{i_t,t}(p_{i_t,t}; z_{i_t,t})$. Update $\mathcal{T}_{i_t,t} = \mathcal{T}_{i_t,t-1} \cup \{t\}$ and $V_{i_t,t} = V_{i_t,t-1} + u_t u'_t$.

8: **end for**

We briefly discuss each step of the algorithm and the intuition behind the theoretical performance. For Steps 1 and 2, the main purpose is to identify the correct neighborhood of the product searched in period t; i.e., $\hat{\mathcal{N}}_{i_t,t} = \mathcal{N}_{i_t}$ with high probability (for brevity of

notation, we let $\hat{\mathcal{N}}_t := \hat{\mathcal{N}}_{i_t,t}$). To achieve that, two conditions are necessary. First, the estimator $\hat{\theta}_{i,t}$ should converge to θ_i as t grows for all $i \in [n]$. Second, the confidence bound $B_{i,t}$ should converge to 0 as t grows, such that in Step 2, we are able to identify different neighborhood by the γ-gap assumption among clusters. To satisfy these conditions, classical statistical learning theory (see e.g., Lemma 2.6.2 in the supplement) requires the minimum eigenvalue of the empirical Fisher's information matrix $V_{i,t}$ to be sufficiently above zero, or more specifically, $\lambda_{\min}(V_{i,t}) \geq \Omega(q_i\sqrt{t})$ (see Lemma 2.6.4 in the supplement). This requirement is guaranteed by the stochastic assumption on demand covariates $z_{i,t}$, which will be imposed in Assumption A-3 in the next subsection, plus our choice of price perturbation in Step 4.

Following the discussion above, when $\hat{\mathcal{N}}_t = \mathcal{N}_{i_t}$ with high probability, we can cluster the data within $\mathcal{N}_{i_t}$ to increase the number of samples for i_t. Because of the increased data samples, it is expected that the estimator $\tilde{\theta}_{\mathcal{N}_{i_t},t-1}$ for θ_{i_t} in Step 3 is more accurate than $\hat{\theta}_{i,t-1}$. Of course, the estimation accuracy again requires the minimum eigenvalue of the empirical Fisher's information matrix over the clustered set $\tilde{\mathcal{T}}_{\mathcal{N}_{i_t},t-1}$, i.e., $\lambda_{\min}(I + \sum_{s\in\tilde{\mathcal{T}}_{\mathcal{N}_{i_t},t-1}} u_s u_s')$, to be sufficiently large, which is again guaranteed by stochastic assumption of $z_{i,t}$ and the price perturbation in Step 4.

The design of the CSMP algorithm depends critically on two things. First, by taking an appropriate price perturbation in Step 4, we balance the exploration and exploitation. If the perturbation is too much, even though it helps to achieve good parameter estimation, it may lead to loss of revenue (due to purposely charging the wrong price). Second, the sequence of demand covariates $z_{i,t}$ has to satisfy an important stochastic assumption (Assumption A-3) which is commonly seen in the pricing literature with demand covariates (see e.g., Chen et al. 2015b, Qiang and Bayati 2016, Ban and Keskin 2017, Javanmard and Nazerzadeh 2019). In the next section, we will drop the stochastic assumption by focusing on a special class of the generalized linear model, the linear demand model, in which the covariates z_t can be non-stochastic or even adversarial.

2.3.2 Theoretical performance of the CSMP algorithm

This section presents the regret of the CSMP pricing policy. Before proceeding to the main result, we first make some technical assumptions that will be needed for the theorem.

Assumption A:

1. The expected revenue function $p\mu(\alpha' x + \beta p)$ has a unique maximizer $p^*(\alpha' x, \beta) \in [\underline{p}, \overline{p}]$, which is Lipschitz in $(\alpha' x, \beta)$ with parameter L_0 for all $x \in \mathcal{X}$ and $\theta \in \Theta$.

Moreover, the unique maximizer is in the interior $(\underline{p}, \overline{p})$ for the true θ_i for all $i \in [n]$ and $x \in \mathcal{X}$.

2. $\mu(\cdot)$ is monotonically increasing and twice continuously differentiable in its feasible region. Moreover, for all $x \in \mathcal{X}$, $\theta \in \Theta$ and $p \in [\underline{p}, \overline{p}]$, we have that $\dot{\mu}(\alpha' x + \beta p) \in [l_1, L_1]$, and $|\ddot{\mu}(\alpha' x + \beta p)| \leq L_2$ for some positive constants l_1, L_1, L_2.

3. For each $i \in [n]$ and $t \in \mathcal{T}_{i,T}$, we have $\mathbb{E}[z_{i,t}|\mathcal{F}_{t-1}] = 0$ and $\lambda_{\min}(\mathbb{E}[z_{i,t} z_{i,t}'|\mathcal{F}_{t-1}]) \geq \lambda_0$ for some $\lambda_0 > 0$, where $\mathcal{F}_{t-1}$ is the σ-algebra generated by history (for instance, $\{i_s, z_s, p_s, d_{i_s,s} : s \leq t-1\}$) until end of period $t-1$.

The first assumption A-1 is a standard regularity condition on expected revenue, which is prevalent in the pricing literature (see e.g., Broder and Rusmevichientong 2012). The second assumption A-2 states that the purchasing probability will increase if and only if the utility $\alpha' x + \beta p$ increases, which is plausible. One can easily verify that the commonly used demand models, such as linear and logistic demand, satisfy these two assumptions with appropriate choice of $\mathcal{X}$ and Θ. The last assumption A-3 is a standard stochastic assumption on demand covariates which has appeared in several pricing papers (see e.g., Qiang and Bayati 2016, Ban and Keskin 2017, Nambiar et al. 2018, Javanmard and Nazerzadeh 2019). In Section 2.4, we will relax this stochastic assumption in the setting of linear demand. Note that A-3 does not require the feature sequence $z_{i,t}$ to be independent or identically distributed, and only requires it to be an adapted sequence of filtration $\{\mathcal{F}_s\}_{s \geq 0}$. One may argue that there can be static or nearly static features in $z_{i,t}$ such that $\lambda_{\min}(\mathbb{E}[z_{i,t} z_{i,t}'|\mathcal{F}_{t-1}]) \geq \lambda_0 > 0$ is violated. However, such static features can be removed from $z_{i,t}$ since the utility corresponding to these static features can be in the constant term, i.e., the intercept in $\alpha_{i_t}'(1, z_{i,t})$. We will see in the numerical study in Section 2.5.1 that our algorithm performs well even when some features are nearly static or slowly changing.

Under Assumption A, we have the following result on the regret of the CSMP algorithm.

Theorem 2.3.1 *Let input parameter $c \geq 20/l_1^2$; the expected regret of algorithm CSMP is*

$$R(T) = O\left(\frac{d^2 \log^2(dT)}{\min_{i \in [n]} q_i^2} + d\sqrt{mT} \log T\right). \tag{2.9}$$

In particular, if $q_i = \Theta(1/n)$ for all $i \in [n]$ and we hide the logarithmic terms, then when $T \gg n$, the expected regret is at most $\tilde{O}(d\sqrt{mT})$.

Sketch of proof. For ease of presentation and to highlight the main idea, we only provide

a proof sketch for the "simplified" regret $\tilde{O}(d\sqrt{mT})$. The proof of the general case (2.9) is given in the supplement.

We show that there is a time threshold $\bar{t} = O(d^2 \log^2(dT)/\min_{i\in[n]} q_i^2)$ such that for all $t > 2\bar{t}$, with high probability we will have $\hat{\mathcal{N}}_t = \mathcal{N}_{i_t}$ (see Lemma 2.6.5 in the supplement). This shows that parameters are accurately estimated when t is sufficiently large, which leads to the desired regret. While for $t \le 2\bar{t}$, the regret can be bounded by $O(\bar{t})$, which is only poly-logarithmic in T and n. To provide a more detailed argument, we first define $\tilde{q}_j := \sum_{i\in\mathcal{N}_j} q_i$ as the probability that a customer views a product belonging to cluster j, and $\tilde{\theta}_{j,t-1} := \tilde{\theta}_{\mathcal{N}_j,t-1}$ as the estimated parameter of cluster j using data in $\tilde{\mathcal{T}}_{j,t-1} := \bigcup_{i\in\mathcal{N}_j} \mathcal{T}_{i,t-1}$, and define $\tilde{T}_{j,t-1} := |\tilde{\mathcal{T}}_{j,t-1}|$. Then, we define

$$\begin{aligned}
\mathcal{E}_{N,t} :=& \{\hat{\mathcal{N}}_t = \mathcal{N}_{i_t}\},\\
\mathcal{E}_{B_j,t} :=& \{||\tilde{\theta}_{j,t} - \theta_j||_2 \le \tilde{B}_{j,t}\},\\
\mathcal{E}_{V,t} :=& \left\{\lambda_{\min}\left(\sum_{s\in\tilde{\mathcal{T}}_{j_t,t}} u_s u_s'\right) \ge \frac{\lambda_1 \Delta_0^2 \sqrt{\tilde{q}_{j_t} t}}{8}\right\},
\end{aligned}$$

where $\lambda_1 = \min(1,\lambda_0)/(1+\overline{p}^2)$ is some constant. Moreover, define

$$\tilde{B}_{j,t} =: \frac{\sqrt{c(d+2)\log(1+t))}}{\sqrt{\lambda_{\min}(\tilde{V}_{j,t})}},$$

where $\tilde{V}_{j,t} = I + \sum_{s\in\tilde{\mathcal{T}}_{j,t}} u_s u_s'$. We further define the event

$$\mathcal{E}_t := \bigcup_{j\in[m]} \mathcal{E}_{B_j,t} \cup \mathcal{E}_{N,t} \cup \mathcal{E}_{V,t}.$$

In the supplement, we will show that $\mathcal{E}_t$ holds with probability at least $1 - 10n/t$ when $t > 2\bar{t}$. So the regret on the event that $\mathcal{E}_t$ fails is at most $O(n\log T)$ because

$$\sum_{t=1}^{T} \mathbb{E}[(r_t(p_t^*) - r_t(p_t))\mathbf{1}(\bar{\mathcal{E}}_t)] \le \overline{p}\sum_{t=1}^{T} \mathbb{P}(\bar{\mathcal{E}}_t) \le 10\overline{p}n\sum_{t=1}^{T} 1/t = O(n\log T).$$

We bound the regret for each period on $\mathcal{E}_t$ as follows. On the event $\mathcal{E}_t$, we apply Taylor's theorem (note that p_t^* is the interior point within the price bound), that under the event $\mathcal{E}_t$ and Assumption A (see also the derivation of (2.20) in the supplement):

$$\mathbb{E}\left[r_t(p_t^*) - r_t(p_t)\right] \le O\left(\mathbb{E}\left[\tilde{B}_{j_t,t-1}^2 + \Delta_t^2\right]\right) \tag{2.10}$$

where

$$\Delta_t = \Delta_{i_t,t}$$

for the sake of brevity. By plugging the value of $\tilde{B}_{j_t,t}$ (with the lower bound of $\lambda_{\min}(\tilde{V}_{j_t,t})$ on the event $\mathcal{E}_{V,t}$), we obtain

$$\begin{aligned}\sum_{t>2\bar{t}} \mathbb{E}\left[\tilde{B}^2_{j_t,t-1}\right] &\le O(d\log T)\sum_{t>2\bar{t}} \mathbb{E}\left[\frac{1}{\sqrt{\tilde{q}_{j_t} t}}\right] \\ &\le O(d\log T)\sum_{t>2\bar{t}}\sum_{j\in[m]} \frac{\sqrt{\tilde{q}_j}}{\sqrt{t}} \\ &\le O(d\log T)\sum_{j\in[m]} \sqrt{\tilde{q}_j T} \le O(d\log T)\sqrt{mT},\end{aligned} \tag{2.11}$$

where the first inequality follows from the definition of $\tilde{B}_{j_t,t-1}$ and event $\mathcal{E}_{V,t}$, the second inequality is from realizations of j_t (i.e., $j_t = j$ with probability $\tilde{q}_j$ for all $j \in [m]$), and the last inequality is by Cauchy-Schwarz.

On the other hand, because $\hat{\mathcal{N}}_t = \mathcal{N}_{i_t}$ for all $t > 2\bar{t}$, we have

$$\mathbb{E}\left[\sum_{t>2\bar{t}} \Delta_t^2\right] \le \sum_{j\in[m]} \mathbb{E}\left[\sum_{t\in\tilde{\mathcal{T}}_{j,T}} \frac{\Delta_0^2}{\sqrt{\tilde{T}_{j,t}}}\right] \le O\left(\mathbb{E}\left[\sum_{j\in[m]} \sqrt{\tilde{T}_{j,T}}\right]\right) \le O\left(\sqrt{mT}\right), \tag{2.12}$$

where the first inequality follows from definition of Δ_t and the event $\hat{\mathcal{N}}_t = \mathcal{N}_{i_t}$.

Putting (2.10), (2.11), and (2.12) together, we obtain

$$\sum_{t\ge 2\bar{t}} \mathbb{E}[r_t(p_t^*) - r_t(p_t)] \le O(d\log T\sqrt{mT}).$$

Thus, the result is proved.

We have a number of remarks about the CSMP algorithm and the result on regret, following in order.

Remark 2.3.1 ***(Comparison with single-product pricing)*** *Our pricing policy achieves the regret $\tilde{O}(d\sqrt{mT})$. A question arises as to how it compares with the baseline single-product pricing algorithm that treats each product separately. Ban and Keskin (2017) consider a single-product pricing problem with demand covariates. According to Theorem 2 in Ban and Keskin (2017), their algorithm, when applied to each product i in our setting separately, achieves the regret $\tilde{O}(d\sqrt{T_{i,T}})$. Therefore, adding together all products $i \in [n]$, the upper bound of the total regret is $\tilde{O}(d\sqrt{nT})$. When the number of clusters m is*

much smaller than n, the regret $\tilde{O}(d\sqrt{mT})$ of CSMP significantly improves the total regret obtained by treating each product separately.

Remark 2.3.2 ***(Lower bound of regret)*** *To obtain a lower bound for the regret of our problem, we consider a special case of our model in which the decision maker knows the underlying true clusters $\mathcal{N}_j$. Since this is a special case of our problem (which is equivalent to single-product pricing for each cluster $\mathcal{N}_j$), the regret lower bound of this problem applies to ours as well. Theorem 1 in Ban and Keskin (2017) shows that the regret lower bound for each cluster j has to be at least $\Omega\left(d\sqrt{\tilde{T}_{j,t}}\right)$. In the case that $\tilde{q}_j = 1/m$ for all $j \in [m]$, it can be derived that the regret lower bound for all clusters has to be at least $\Omega(d\sqrt{mT})$. This implies that the regret of the proposed CSMP policy is optimal up to a logarithmic factor.*

Remark 2.3.3 ***(Improving the regret for large n)*** *When n is large, the first term in our regret bound $O(d^2 \log^2(dT)/\min_{i\in[n]} q_i^2)$ will also become large. For instance, if $q_i = O(1/n)$ for all $i \in [n]$, then this term becomes $O(d^2 n^2 \log^2(dT))$. One way to improve the regret, although it requires prior knowledge of γ, is to conduct more price exploration during the early stages. Specifically, if the confidence bound $B_{i,t-1}$ of product i is larger than $\gamma/4$, in Step 4, we let the price perturbation $\Delta_{i,t}$ be $\pm\Delta_0$ to introduce sufficient price variation (otherwise let Δ_t be the same as in the original algorithm CSMP). Following a similar argument as in Lemma 2.6.4 in the supplement, it roughly takes $O(d\log(dT)/\min_{i\in[n]} q_i)$ time periods before all $B_{i,t-1} < \gamma/4$, so the same proof used in Theorem 2.3.1 applies. Therefore, when $q_i = O(1/n)$ for all $i \in [n]$, the final regret upper bound is $O(dn\log(dT) + d\log T\sqrt{mT})$.*

Remark 2.3.4 ***(Relaxing the cluster assumption)*** *Our theoretical development assumes that products within the same cluster have exactly the same parameters θ_i. This assumption can be relaxed as follows. Define two products i_1, i_2 as in the same cluster if they satisfy $||\theta_{i_1} - \theta_{i_2}||_2 \leq \gamma_0$ for some positive constant γ_0 with $\gamma_0 < \gamma/2$ (as earlier, otherwise they satisfy $||\theta_{i_1} - \theta_{i_2}||_2 > \gamma$). Our policy in Algorithm 1 can adapt to this case by modifying Step 2 to*

$$\hat{\mathcal{N}}_{i,t} = \{i' \in [n] : \ ||\hat{\theta}_{i',t-1} - \hat{\theta}_{i,t-1}||_2 \leq B_{i',t-1} + B_{i,t-1} + \gamma_0\},$$

and we let

$$\Delta_{i,t} = \pm\Delta_0 \max\left(\hat{T}_{\hat{\mathcal{N}}_{i,t},t}^{-1/4}, \upsilon\right),$$

where $\upsilon = \Theta(\gamma_0^{1/3})$ is a constant. Following almost the same analysis, we can show that the regret is at most $\tilde{O}(d\sqrt{mT} + \gamma_0^{2/3}T)$. We refer the interested reader to Theorem 2.6.1 in the supplement for a more detailed discussion. The main difference between this regret and the one obtained in Theorem 2.3.1 is the extra term $\tilde{O}(\gamma_0^{2/3}T)$. It is clear that when $\gamma_0 = 0$, we have exactly the same regret as in Theorem 2.3.1. In general, if γ_0 is small (e.g., in the order of $T^{-3/4}$), then $\tilde{O}(d\sqrt{mT} + \gamma_0^{2/3}T)$ can still be a better regret than $\tilde{O}(d\sqrt{nT})$, which is the typical regret of single-product pricing problems for n products. As a result, the idea of clustering can be useful even if the parameters within the same cluster are different.

2.4 Pricing Policy for Linear Model

The previous sections developed an adaptive policy for a generalized linear demand model under a stochastic assumption on the covariates z_t. This assumption may be too strong in some applications. As argued in some of the adversarial bandit literature, some terms in the reward function may not satisfy any stochastic distribution and can even appear adversarially. In our model, the contextual covariate usually includes such information as customer rating of the product, competitor's price of similar products, promotion information, and average demand of the product in the past few weeks, etc., which may not follow any probability distribution.

In this section, we drop the stochastic assumption by focusing on the linear demand model, which is an important and widely adopted special case of the generalized linear demand model. With a linear demand function, the expected value in (2.1) with covariates $x'_{i,t} = (1, z_{i,t})'$ takes the form

$$\mu(\alpha'_i x_{i,t} + \beta_i p_{i,t}) = \alpha'_i x_{i,t} + \beta_i p_{i,t}. \tag{2.13}$$

We point out that (2.13) is interpreted as purchasing probability in the previous section when each period has a single customer. The linear demand model typically applies when the demand size in period t is random and given by

$$d_{i,t}(x_{i,t}, p_{i,t}) = \alpha'_i x_{i,t} + \beta_i p_{i,t} + \epsilon_{i,t},$$

where $\epsilon_{i,t}$ is a zero-mean and sub-Gaussian random variable. Then (2.13) represents the average demand in period t. While our pricing policy applies to both cases, we focus on the case that (2.13) represents purchasing probability for the consistency and simplicity of presentation.

For the linear demand model, we can relax Assumption A to the following.

Assumption B:

1. There exists some compact interval of negative numbers $\mathcal{B}$, such that $\beta_i \in \mathcal{B}$ for each $i \in [n]$, and $-\alpha_i' x/(2\beta_i) \in (\underline{p}, \overline{p})$ for all $x \in \mathcal{X}$.

2. For any $i \in [n]$ and $t \in [T]$ such that $T_{i,t} \geq t_0$, $\lambda_{\min}(\sum_{s \in \mathcal{T}_{i,t}} x_s x_s') \geq c_0 T_{i,t}^{\kappa}$ for some constant $c_0, t_0 > 0$ and $\kappa \in (1/2, 1]$.

We note that, compared with Assumption A, the first two assumptions in Assumption A are automatically satisfied for the linear demand model with Assumption B-1. The condition $\beta_i < 0$ is natural since β_i is the coefficient of the price sensitivity in (2.13). Essentially, Assumption B-2 relaxes the stochastic assumption on demand covariates in Assumption A-3 such that covariates can be chosen arbitrarily as long as they have enough "variation". The reasons that Assumption B-2 is a relaxation of Assumption A-3 are the following. First, as mentioned earlier, the covariates may not follow any distribution at all. Second, one can verify that if Assumption A-3 is satisfied, then Assumption B-2 is also satisfied with probability at least $1 - \Delta$ (for any $\Delta > 0$) given $t_0 = O(\log(dn/\Delta))$, $c_0 = 1/2$, and $\kappa = 1$ (according to the proof in Lemma 2.6.4 in the supplement). Third, in real application, Assumption A-3 is difficult to verify, while Assumption B-2 can be verified from the data by simply observing the historical demand covariates of each product. Finally, we point out that Assumption B-2 is needed only for identifying clusters of products, so it is not necessary and can be dropped for the single-product pricing problem.

For linear demand, we are able to separately estimate α_i and β_i. First, it can be shown that β_i can be estimated accurately using a simple estimation approach below. Then, α_i can be easily estimated using a regularized linear regression (e.g., ridge regression). To guarantee accurate parameter estimation for α_i, classical regression theory requires the minimum eigenvalue of empirical Fisher's information matrix to be sufficiently large. With α_i estimated separately from β_i, its empirical Fisher's information matrix is $\bar{V}_{i,t} := I + \sum_{s \in \mathcal{T}_{i,t}} x_s x_s'$. This explains why Assumption B-2 on $\bar{V}_{i,t}$, instead of the stochastic assumption on demand covariates A-3 for the GLM case, is required for the linear demand model.

To conduct separate parameter estimation, we adopt the idea from Nambiar et al. (2018). Let

$$\hat{\beta}_{i,t} := \text{Proj}_{\mathcal{B}} \left(\frac{\sum_{s \in \mathcal{T}_{i,t}} \Delta_s d_s}{\sum_{s \in \mathcal{T}_{i,t}} \Delta_s^2} \right) \tag{2.14}$$

be the estimated parameter of β_i using individual data in $\mathcal{T}_{i,t}$. We will show that under certain conditions, $\hat{\beta}_{i,t}$ is an accurate estimation of β_i. To estimate α_i, we apply the idea of

regularization. That is,

$$\hat{\alpha}_{i,t} = \arg\min \sum_{s \in \mathcal{T}_{i,t}} (d_s - \alpha' x_s - \hat{\beta}_{i,t} p_s)^2 + \lambda_\alpha ||\alpha||_2^2. \tag{2.15}$$

We notice that when $\hat{\beta}_{i,t}$ is sufficiently close to β_i, $\hat{\alpha}_{i,t}$ is essentially a ridge regression estimator of α_i, whose estimation error is well-studied (see, e.g., Abbasi-Yadkori et al. 2011). To simplify our presentation, in what follows we set the ℓ_2 regularization parameter λ_α in (2.15) as 1. From our numerical studies, we observe that the performance is not sensitive to the choice of λ_α when T is large. Similarly, using clustered data from $\tilde{\mathcal{T}}_{\hat{\mathcal{N}}_{i,t},t}$, we can obtain the estimators $\tilde{\beta}_{\hat{\mathcal{N}}_{i,t},t}$ and $\tilde{\alpha}_{\hat{\mathcal{N}}_{i,t},t}$.

We refer to our algorithm in this section as Clustered Semi-Myopic Pricing for Linear model (CSMP-L), which is presented in Algorithm 2. The structure of CSMP-L is similar to CSMP in Algorithm 1. The main difference is that CSMP-L constructs different confidence bounds to determine the neighborhood $\hat{\mathcal{N}}_{i,t}$ of product i. In particular, in Step 3 in Algorithm 2, we define

$$C_{i,t} = \sqrt{(\tilde{C}_{i,t}^{\beta})^2 + (\tilde{C}_{i,t}^{\alpha})^2 / \lambda_{\min}(\bar{V}_{i,t})}, \tag{2.16}$$

where

$$\begin{aligned} \tilde{C}_{i,t}^{\beta} =& c_1 \sqrt{\log t} \left(\sum_{s \in \mathcal{T}_{i,t}} \Delta_s^2 \right)^{-1/2}, \\ \tilde{C}_{i,t}^{\alpha} =& c_2 \sqrt{(d+1)} \log t \left(\sum_{s \in \mathcal{T}_{i,t}} \Delta_s^2 \right)^{-1/2} \sqrt{T_{i,t}}, \end{aligned} \tag{2.17}$$

for some constant $c_1 > 0, c_2 > 0$. The choice of c_1 and c_2 will be further discussed in the numerical experiments section.

The next theorem presents the theoretical performance of the CSMP-L algorithm in terms of the regret.

Theorem 2.4.1 *The expected regret of algorithm CSMP-L is*

$$R(T) = O\left(\left(\frac{\sqrt{d} \log T}{\min_{i \in [n]} q_i^{\kappa/2}} \right)^{4/(2\kappa - 1)} + d^2 \sqrt{mT} (\log T)^3 \right). \tag{2.18}$$

If we hide logarithmic terms and suppose $\min_{i \in [n]} q_i = \Theta(1/n)$ *with* $T \gg n$*, the expected*

Algorithm 2 The CSMP-L Algorithm

Require: c_1, c_2, confidence bound parameters; Δ_0, price perturbation parameter;

1: **Step 0. Initialization.** Initialize $\mathcal{T}_{i,0} = \emptyset$ and $\bar{V}_{i,0} = I$ for all $i \in [n]$. Let $t = 1$, go to Step 1.

2: **for** $t = 1, 2, \ldots, T$ **do**

3: **Step 1. Individual Parametric Estimation.** Compute the estimated parameters $\hat{\theta}'_{i,t-1} = (\hat{\alpha}_{i,t-1}, \hat{\beta}_{i,t-1})$ for all $i \in [n]$ as

$$\hat{\beta}_{i,t-1} = \text{Proj}_{\mathcal{B}} \left(\frac{\sum_{s \in \mathcal{T}_{i,t-1}} \Delta_s d_s}{\sum_{s \in \mathcal{T}_{i,t-1}} \Delta_s^2} \right)$$

and

$$\hat{\alpha}_{i,t-1} = \arg\min \sum_{s \in \mathcal{T}_{i,t-1}} (d_s - \alpha' x_s - \hat{\beta}_{i,t-1} p_s)^2 + ||\alpha||_2^2.$$

Go to Step 2.

4: **Step 2. Estimating Neighborhood.** Compute the neighborhood of i as

$$\hat{\mathcal{N}}_{i,t} = \{i' \in [n] : \ ||\hat{\theta}_{i',t-1} - \hat{\theta}_{i,t-1}||_2 \leq C_{i',t-1} + C_{i,t-1}\}$$

where $C_{i,t-1}$ is defined in (2.16) for all $i \in [n]$. Go to Step 3.

5: **Step 3. Clustered Parametric Estimation.** Compute the estimated parameter $\tilde{\theta}'_{\hat{\mathcal{N}}_{i,t},t-1} = (\tilde{\alpha}'_{\hat{\mathcal{N}}_{i,t},t-1}, \tilde{\beta}_{\hat{\mathcal{N}}_{i,t},t-1})$ using clustered data

$$\tilde{\beta}_{\hat{\mathcal{N}}_{i,t},t-1} = \text{Proj}_{\mathcal{B}} \left(\frac{\sum_{s \in \tilde{\mathcal{T}}_{\hat{\mathcal{N}}_{i,t},t-1}} \Delta_s d_s}{\sum_{s \in \tilde{\mathcal{T}}_{\hat{\mathcal{N}}_{i,t},t-1}} \Delta_s^2} \right)$$

and

$$\tilde{\alpha}_{\hat{\mathcal{N}}_{i,t},t-1} = \arg\min \sum_{s \in \tilde{\mathcal{T}}_{\hat{\mathcal{N}}_{i,t},t-1}} (d_s - \alpha' x_s - \tilde{\beta}_{\hat{\mathcal{N}}_{i,t},t-1} p_s)^2 + ||\alpha||_2^2.$$

for each $i \in [n]$. Go to Step 4.

6: **Step 4. Pricing.** Compute price for each $i \in [n]$ as

$$p'_{i,t} = \arg\max_{p \in [\underline{p}, \bar{p}]} (\tilde{\alpha}'_{\hat{\mathcal{N}}_{i,t},t-1} x_{i,t} + \tilde{\beta}_{\hat{\mathcal{N}}_{i,t},t-1} p) p,$$

then project to $\tilde{p}_{i,t} = \text{Proj}_{[\underline{p}+|\Delta_{i,t}|, \bar{p}-|\Delta_{i,t}|]}(p'_{i,t})$ and offer to the customer price $p_{i,t} = \tilde{p}_{i,t} + \Delta_{i,t}$ where $\Delta_{i,t} = \pm \Delta_0 \tilde{T}_{\hat{\mathcal{N}}_{i,t},t}^{-1/4}$ which takes two signs with equal probability.

7: Then, customer in period t searchers for product i_t, and makes purchase decision $d_{i_t,t}(p_{i_t,t}; z_{i_t,t})$, and update $\mathcal{T}_{i_t,t} = \mathcal{T}_{i_t,t-1} \cup \{t\}$ and $\bar{V}_{i_t,t} = \bar{V}_{i_t,t-1} + x_t x'_t$.

8: **end for**

regret is at most $\tilde{O}(d^2\sqrt{mT})$.

Compared with Theorem 2.3.1, it is seen that the regret of CSMP-L is slightly worse than that of CSMP by the dimension d and some logarithmic terms. This is attributed to the weakened assumption on covariate vectors. However, in contrast to Theorem 2.3.1 where the regret is taken over the expectation with regard to the stochastic feature $z_t, t \in [T]$, the regret in (2.18) holds for any feature vector, even when the feature vectors $z_t, t \in [T]$, are chosen adversarially.

Remark 2.4.1 *Assumption B-2 (for linear model) and Assumption A-3 (for generalized linear model) require the product features to have sufficient variations. These two assumptions are made only for the purpose of identifying product clusters. That is, if the clustering of products is known a priori, e.g., the single-product dynamic pricing problem, then these assumptions can be completely dropped (i.e.,* z_t *can be chosen completely arbitrarily), and the results continue to hold. We offer a justification for making this assumption. By our definition of cluster, we need* $\mathbb{E}[||\hat{\theta}_{i,t} - \theta_i||_2] \leq \gamma$ *to identify the right cluster for product* i*. On the other hand, classic statistics theory (e.g., Cramér-Rao lower bound) states that* $\mathbb{E}[||\hat{\theta}_{i,t} - \theta_i||_2] \geq \Omega(1/\sqrt{\lambda_{\min}(V_{i,t})})$*. Therefore, if the product features do not have sufficient variation, it is essentially not possible to have the estimation error bounded above by* γ *to find the right cluster for* i*.*

2.5 Simulation Results and Field Experiments

This section provides the simulation and field experiment results for algorithms CSMP and CSMP-L. First, we conduct a simulation study using synthetic data in Section 2.5.1 to illustrate the effectiveness and robustness of our algorithms against several benchmark approaches. Second, the simulation results using a real dataset from Alibaba are provided in Section 2.5.2. Third, Section 2.5.3 reports the results from a field experiment at Alibaba. Finally, we summarize all numerical experiment results in Section 2.5.4.

2.5.1 Simulation using synthetic data

In this section, we demonstrate the effectiveness of our algorithms using some synthetic data simulation. We first show the performance of CSMP and CSMP-L against several benchmark algorithms. Then, several robustness tests are conducted for CSMP. The first test is for the case when clustering assumption is violated (i.e., parameters within the same cluster are slightly different). The second test is when the demand covariates $z_{i,t}$ contain

some features that change slowly in a deterministic manner. Finally, we test CSMP with a misspecified demand model.

We shall compare the performance of our algorithms with the following benchmarks:

- The Semi-Myopic Pricing (SMP) algorithm, which treats each product independently (IND), and we refer to it as SMP-IND.
- The Semi-Myopic Pricing (SMP) algorithm, which treats all products as one (ONE) single cluster, and we refer to the algorithm as SMP-ONE.
- The Clustered Semi-Myopic Pricing with K-means Clustering (CSMP-KMeans), which uses K-means clustering for product clustering in Step 2 of CSMP.

The first two benchmarks are natural special cases of our algorithm. Algorithm SMP-IND skips the clustering step in our algorithm and always sets the neighborhood as $\hat{\mathcal{N}}_t = \{i_t\}$; while SMP-ONE keeps $\hat{\mathcal{N}}_t = \mathcal{N}$ for all $t \in [T]$. The last benchmark is to test the effectiveness of other classical clustering approach for our setting, in which we choose K-means clustering as an illustrative example because of its popularity.

Logistic demand with clusters. We first simulate the demand using a logistic function. We set the time horizon $T = 30,000$, the searching probability $q_i = 1/n$ for all $i \in [n]$ where $n = 100$, and the price range $\underline{p} = 0$ and $\overline{p} = 10$. In this study, it is assumed that all $n = 100$ products have $m = 10$ clusters (with products randomly assigned to clusters). Within a cluster j, each entry in α_j is generated uniformly from $[-L/\sqrt{d+2}, L/\sqrt{d+2}]$ with $L = 10$, and β_j is generated uniformly from $[-L/\sqrt{d+2}, 0)$ (to guarantee that $||\theta_i||_2 \leq L$). For demand covariates, each feature in $z_{i,t}$, with dimension $d = 5$, is generated independently and uniformly from $[-1/\sqrt{d}, 1/\sqrt{d}]$ (to guarantee that $||z_{i,t}||_2 \leq 1$). For the parameters in the algorithms, we let $\Delta_0 = 1$; and for the confidence bound $B_{i,t} = \sqrt{c(d+2)\log(1+t)/\lambda_{\min}(V_{i,t})}$, we first let $c = 0.8$ and then test other values of c for sensitivity analysis. For the benchmark CSMP-KMeans, we need to specify the number of clusters K; since the true number of clusters m is not known *a priori*, we test different values of K in $\{5, 10, 20, 30\}$. Note that when $K = 10$, the performance of CSMP-KMeans can be considered as an oracle since it correctly specifies the true number of product clusters.

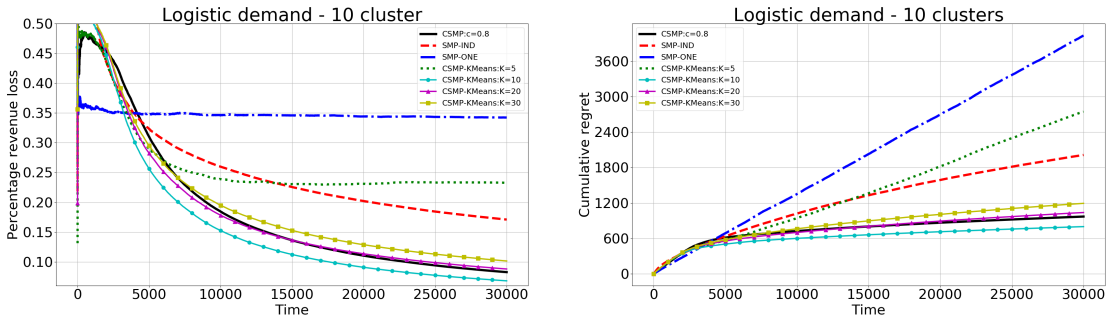
To evaluate the performance of algorithms, we adopt both the cumulative regret in (2.4) and the percentage revenue loss defined by

$$L^{\pi}(T) = \frac{R^{\pi}(T)}{\sum_{t=1}^{T} \mathbb{E}[r_t(p_t^*)]}, \tag{2.19}$$

which measures the percentage of revenue loss with respect to the optimal revenue. Obviously, the percentage revenue loss and cumulative regret are equivalent, and a better policy leads to a smaller regret and a smaller percentage revenue loss.

For each experiment, we conduct 30 independent runs and take their average as the output. We also output the standard deviation of percentage revenue loss for all policies in Table 2.1. It can be seen that our policy CSMP has quite small standard deviation, so we will neglect standard deviation results in other experiments.

We recognize that a more appropriate measure for evaluating an algorithm is the regret (and percentage of loss) of expected total profit (instead of expected total revenue). We choose the latter for the following reasons. First, it is consistent with the objective of this chapter, which is the choice of the existing literature. Second, it is revenue, not profit, that is being evaluated at our industry partner, Alibaba. Third, even if we wish to measure it using profit, the cost data of products are not available to us, since the true costs depend on such critical things as terms of contracts with suppliers, that are confidential information.



(a) Plot of percentage revenue loss

(b) Plot of cumulative regret

Figure 2.2: Performance of different policies for logistic demand with 10 clusters.

The results are shown in Figure 2.2. The graph on the left-hand side shows the percentage revenue loss of all algorithms, and the graph on the right-hand side shows the cumulative regrets for each algorithm. The black solid line represents CSMP, the red dashed line represents SMP-IND, the blue dash-dotted line represents SMP-ONE, the green dotted line represents CSMP-KMeans with $K = 5$, the cyan solid line with round marks represents CSMP-KMeans with $K = 10$, the purple solid line with triangle marks represents CSMP-KMeans with $K = 20$, and the yellow solid line with square marks represents CSMP-KMeans with $K = 30$. According to this figure, our algorithm CSMP outperforms all the benchmarks except for CSMP-KMeans when $K = m = 10$. CSMP-KMeans with $K = 10$ has the best performance, which is not surprising because it uses the exact and correct number of clusters. However, in reality the true cluster number m is not known. We

	$t = 5,000$	$t = 10,000$	$t = 15,000$	$t = 20,000$	$t = 25,000$	$t = 30,000$
CSMP	1.83	0.97	0.70	0.57	0.47	0.40
SMP-IND	1.32	0.88	0.92	0.81	0.78	0.73
SMP-ONE	2.34	2.15	1.75	1.44	1.46	1.44
CSMP-KMeans:$K = 5$	2.08	1.97	1.95	2.26	2.22	2.19
CSMP-KMeans:$K = 10$	2.06	1.53	1.09	0.87	0.74	0.66
CSMP-KMeans:$K = 20$	2.12	1.36	1.15	1.02	0.91	0.82
CSMP-KMeans:$K = 30$	1.41	0.88	0.77	0.67	0.59	0.49

Table 2.1: Standard deviation (%) of percentage revenue loss corresponding to different time periods for logistic demand with 10 clusters.

	$c = 0.5$	$c = 0.6$	$c = 0.7$	$c = 0.8$	$c = 0.9$	$c = 1.0$
Mean	8.56	8.28	8.52	8.27	8.56	8.72
Standard deviation	0.73	0.51	0.73	0.40	0.66	0.35

Table 2.2: Mean and standard deviation (%) of percentage revenue loss of CSMP (logistic demand with 10 clusters) with different parameters c.

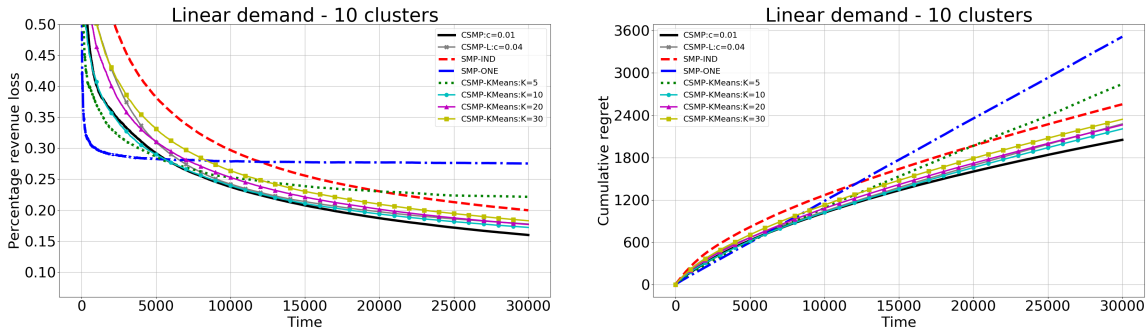
also test CSMP-KMeans with $K = 5, 20, 30$. We find that when $K = 20$, its performance is similar to (slightly worse than) our algorithm CSMP. When $K = 5, 30$, the performance of CSMP-KMeans becomes much worse (especially when $K = 5$). For the other two benchmarks SMP-ONE and SMP-IND, their performances are not satisfactory either, with SMP-ONE has the worst performance because clustering all products together leads to significant error. Sensivitiy results of CSMP with different parameters c are presented in Table 2.2, and it can be seen that CSMP is quite robust with different values of c.

Linear demand with clusters. Now we present the results of CSMP and CSMP-L with linear demand function. For synthetic data, $z_{i,t}$ is generated the same way as in the logistic demand case but with $L = 1$ (in order for the purchasing probability to be within $[0, 1]$), and $n, m, T, q_i, d, \Delta_0$ and price ranges are also kept the same. For demand parameters, $\alpha_{j,k} \in [0, L/\sqrt{d+2}]$ for each entry k corresponding to context z_t, $\alpha_{j,k} \in [L/\sqrt{d+2}, 2L/\sqrt{d+2}]$ for k corresponding to the intercept, and the price sensitivity $\beta_j \in [-1.05L/\sqrt{d+2}, -0.05L/\sqrt{d+2}]$. The reason for this construction of data is to guarantee that the linear purchasing probabilities are mostly within $[0, 1]$. Besides CSMP (with $c = 0.01$), this experiment also tests the algorithm CSMP-L. For input parameters of CSMP-L, the confidence bound $C_{i,t}$ is set to

$$\sqrt{c\left(\log t / \sum_{s\in\mathcal{T}_{i,t}} \Delta_s^2 + 0.05(d+1)\log^2 tT_{i,t}/(\lambda_{\min}(\bar{V}_{i,t}) \sum_{s\in\mathcal{T}_{i,t}} \Delta_s^2)\right)},$$

with $c = 0.04$. The results are summarized in Figure 2.3 (the grey solid line with X marks represents CSMP-L). It can be seen that our algorithm CSMP has the best performance, even exceeding CSMP-KMeans with $K = 10$. The reason might be that since $L = 1$ (instead of $L = 10$ for the logistic demand case), the parameters are closer to each other, hence it becomes more difficult to be clearly separated by K-means method. For algorithm CSMP-L, its numerical performance is slightly worse than CSMP, but still performs better than benchmarks SMP-IND and SMP-ONE.

Since logistic demand is more commonly used to model probability, in the following robustness check of CSMP, we only test logistic demand as an illustration.



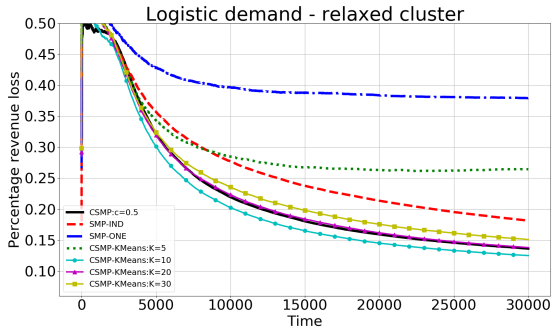
(a) Plot of percentage revenue loss

(b) Plot of cumulative regret

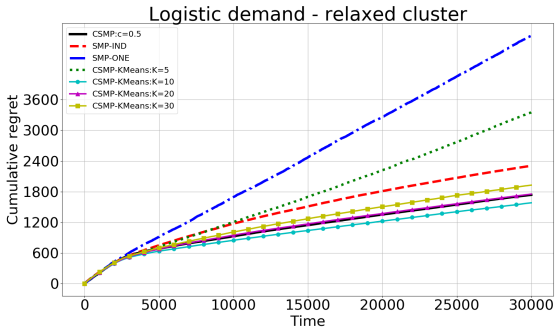
Figure 2.3: Performance of different policies for linear demand with 10 clusters.

Logistic demand with relaxed clusters. As we discussed in Section 2.3.2, strict clustering assumption might not hold and sometimes products within the same cluster are slightly different. This experiment tests the robustness of CSMP when parameters of products in the same cluster are slightly different. To this end, after we generate the $m = 10$ centers of parameters (with each center represented by θ_j), for each product i in the cluster j, we let $\theta_i = \theta_j + \Delta\theta_i$ where $\Delta\theta_i$ is a random vector such that each entry is uniformly drawn from $[-L/(10\sqrt{d+2}), L/(10\sqrt{d+2})]$. All the other parameters are the same as in the case with 10 clusters. Results are summarized in Figure 2.4, and it can be seen that the performances of all algorithms are quite similar as in Figure 2.2.

Logistic demand with almost static features. As we discussed after Assumption A-3, in some applications there might be features that have little variations (nearly static). We next test the robustness of our algorithm CSMP when the feature variations are small. To this end, we assume that one feature in $z_{i,t} \in \mathbb{R}^d$ for each $i \in [n]$ is almost static. More specifically, we let this feature be constantly $1/\sqrt{d}$ for 100 periods, then change to $-1/\sqrt{d}$ for another 100 periods, then switch back to $1/\sqrt{d}$ after 100 periods, and this



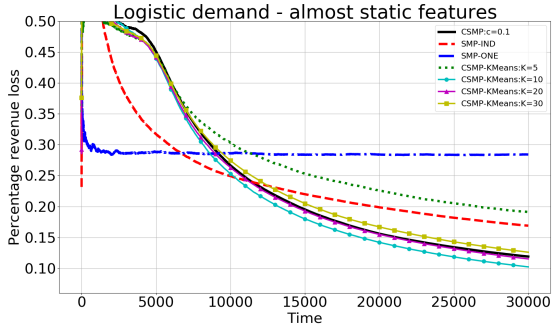
(a) Plot of percentage revenue loss



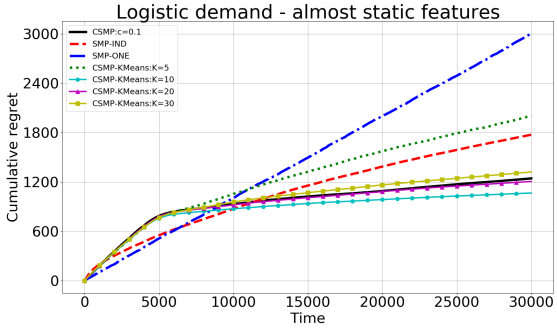
(b) Plot of cumulative regret

Figure 2.4: Performance of different policies for logistic demand with relaxed clusters.

process continues. The numerical results against benchmarks are summarized in Figure 2.5. It can be seen that with such an almost static feature, the performances of algorithms with clustering become worse, but they still outperform the benchmark algorithms. In particular, CSMP (with parameter $c = 0.1$ after a few trials of tuning) still has promising performance, showing its robustness with small feature variations of some products.



(a) Plot of percentage revenue loss



(b) Plot of cumulative regret

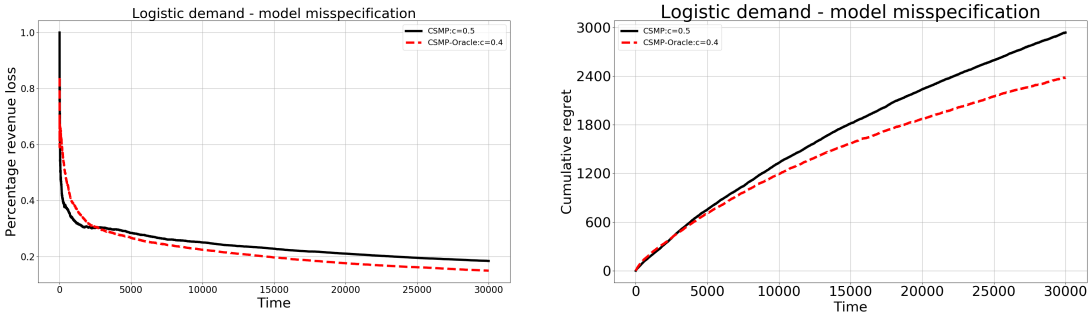
Figure 2.5: Performance of different policies for logistic demand with 10 clusters and almost static features.

Logistic demand with model misspecification. In real applications, it may happen that the demand model is misspecified. In this experiment, we consider a misspecified logistic demand model. Specifically, we let the expected demand of product i be $1/(1+\exp(f_i(z_t, p_t)))$, where the utility function

$$f_i(z_t, p_t) := c_{i,0} + \sum_{k=1}^{d} c_{1,i,k} z_{t,k} + \sum_{k=1}^{d} c_{2,i,k} z_{t,k}^2 + \sum_{k=1}^{d} c_{3,i,k} z_{t,k}^3 + \beta_{1,i} p_t + \beta_{2,i}^2 p_t^2 + \beta_{3,i} p_t^3$$

is a third degree polynomial of z_t, p_t, where c_i, β_i are unknown parameters, and $z_{t,k}$ represents te k-th component of z_t. To generate this misspecified demand model, we let $c_{l,i,k} \in [-L/\sqrt{3(d+2)}, L/\sqrt{3(d+2)}]$ with $l \in \{1,2,3\}, k \in [d]$, $c_{i,0} \in [-L/\sqrt{d+2}, L/\sqrt{d+2}]$, and $\beta_{l,i} \in [-L/\sqrt{3(d+2)}, 0)$ with $l \in \{1,2,3\}$, be all drawn uniformly. All the other input parameters for the problem instance are the same as in the case of logistic demand with 10 clusters.

To test the robustness of the misspecified CSMP, it is compared with CSMP which correctly specifies the demand model. We call the benchmark the CSMP-Oracle. The numerical results are summarized in Figure 2.6. As seen, when compared with the oracle, the misspecified CSMP has slightly worse performance as expected. But the overall difference in percentage revenue loss is only 3.48%, showing that our algorithm CSMP is rather robust with such a model misspecification.



(a) Plot of percentage revenue loss

(b) Plot of cumulative regret

Figure 2.6: Performance of CSMP with (misspecified) logistic demand versus the oracle.

2.5.2 Simulation using real data from Alibaba

This section presents the results of our algorithms (for illustration, we use CSMP with logistic demand) and other benchmarks using a real dataset provided by Alibaba. To better simulate the real demand process, we fit the demand data to create a sophisticated ground truth model (hence our algorithm CSMP may have a model misspecification). Before presenting the results, we introduce the dataset and pre-processing of the data.

The dataset. The dataset is from Tmall Supermarket, which is an online store owned by Alibaba. To motivate our study of pricing for low-sale products, we extract sales data from 05/29/2018 to 07/28/2018. During this period, nearly 75,000 products were offered by Tmall Supermarket. There are more than 21.6% (i.e., 16,000) products with average numbers of daily unique visits less than 10. Among all these low-sale products, Alibaba

provided us with a test dataset comprising 100 products that have at least one sale during the 61-day period, and at least two prices charged with each price offered to more than 10% of all customers. Because these selected products have sufficient variation of prices and different observations of customers' purchases, demand parameters can be estimated quite accurately using the sales data in the dataset.

For the features of products, we are provided by Alibaba with 5 features (hence $d = 5$), that are described below:

- Average gross merchandise volume (GMV, i.e., product revenue) in past 30 days.
- Average demand in past 30 days.
- Average number of unique buyers (UB, i.e., unique IP which makes the purchase) in past 30 days.
- Average number of unique visitors (UV) in past 30 days.
- Average number of independent product views (IPV, i.e., total number of views on the product, including repetitive views from the same user) in past 30 days.

These features are selected by Alibaba's feature engineering team[2] (via a recursive feature elimination approach from a raw set of features). Note that these features are not exogeneous, since features in the future can be affected by current pricing decision. Such endogenous features are often used in the demand forecasting literature. For instance, a time series model uses past demand to predict future demand (see e.g., Brown 1959); an artificial neural network (ANN) model uses historical demand data of composite products as features for demand prediction (Chang et al. 2005). In the pricing literature, some endogenous features have also been used. For example, in Ban and Keskin (2017), Bastani et al. (2019), their model features include auto loan data, e.g., competitors' rate, that are affected by the rate offered by the decision maker (the auto loan company). Incorporating the impact of pricing decisions on features leads to challenging dynamic programming problem with partial information. Hence, features are considered as given and we only optimize for current period (i.e., ignoring the long-run effect of the current pricing decision).

To run simulation using the real dataset, we first create a ground truth model for the demand. We consider two ground truth models in this simulation study. The first one is the commonly used logistic demand function (hence no model misspecification for our

[2]We requested to include some other features, such as number/score of customer ratings and competitor's price on similar product, but were unable to obtain such data due to technical reasons during the field experiment.

algorithm CSMP), and the second is a random forest model (as used in simulation study of Nambiar et al. 2018, hence there is model misspecification for CSMP). We use the demand data of each product to fit these two demand models, and then apply them to simulate the demand process.

We want to generate customer's arrival at each time t, i.e., the product i_t a customer chooses to search. Since the dataset contains the daily number of unique visitors for each product i, the arrival process i_t is simulated by randomly permuting the unique visitors of each product on each day. For instance, if on day 1, product 1 and product 2 have 2 and 3 unique visitors respectively; then i_t for $t = 1, \ldots, 5$ can be $1, 2, 2, 1, 2$, which is a random permutation of the unique visitors for product 1 and 2.

Numerical results for the algorithms. We first provide the specifications of the parameters in the CSMP algorithm in Algorithm 1.

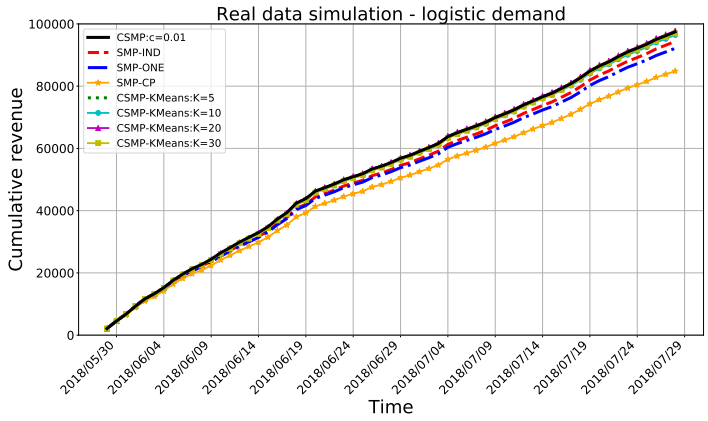
- The confidence bound $B_{i,t}$ is $\sqrt{c(d+2)\log(1+t)/\lambda_{\min}(V_{i,t})}$, where $c = 0.01$ for logistic demand and $c = 0.05$ for random forest demand (selected by a few trials of different values).

- The price lower bound of each product is 50% lower than its lowest price during the 61-day period, and the price upper bound is 50% higher than its highest price during this period of time.

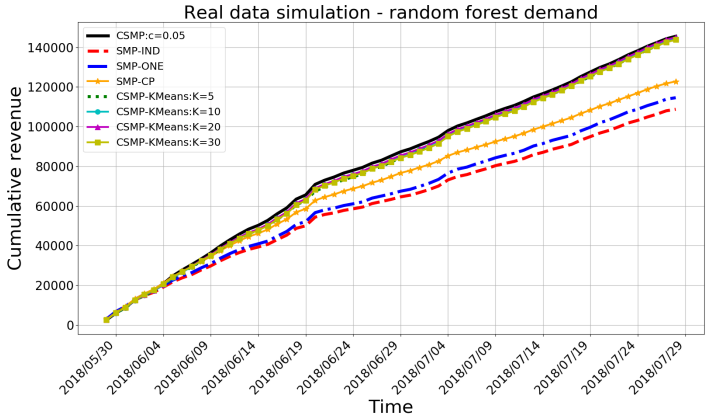
- The basic price perturbation parameter Δ_0 of each product is set as the length of price range divided by 4, i.e., $\Delta_0 = (\overline{p} - \underline{p})/4$.

For benchmark algorithms, they are the same as those in the previous subsection, with CSMP-KMeans have $K \in \{5, 10, 20, 30\}$. In addition, we test another benchmark proposed in Keskin and Zeevi (2016). More specifically, this benchmark assumes a simple linear demand model as $\mathbb{E}[d_{i,t}] = \alpha_{i,t} + \beta'_{i,t} p_{i,t}$ with changing parameters $\alpha_{i,t}, \beta_{i,t}$ but without demand covariates. Since this single-product pricing algorithm can be considered as a modified version of semi-myopic pricing, we call it semi-myopic pricing (SMP) with changing parameters (CP), or SMP-CP for short. We plot the results of cumulative revenue at different dates in Figure 2.7.

It can be seen that all the methods using clustering have better performance, and their performances are comparable. It is interesting to note that for clustering using K-means method, their performances with different value of K are actually quite close. Finally, it is observed that the advantage of using clustering with random forest model (i.e., misspecified model) is more than that with logistic model.



(a) Logistic demand model (without model misspecification)



(b) Random forest demand model (with model misspecification)

Figure 2.7: Plot of cumulative revenue over different dates for two demand models

2.5.3 Field experiment results from Alibaba

We have collaborated with Alibaba Group to implement our algorithm CSMP to a set of products on Tmall Supermarket, and we report some of the findings in this subsection. Due to the privacy policy of Alibaba, some details of the field experiment are not provided.

To conduct the experiment, we randomly selected 390 low-sale products from several categories for our study. Then, 40 products were chosen randomly from them as the testing group and CSMP algorithm were implemented for their pricing decisions, and the rest were used as the control group that continued to use the original pricing policy at Alibaba. Purchasing probability is assumed to be a logistic function, and we use the same input parameters as in Section 2.5.2. We note two implementation details. First, according to the requirement from Alibaba, the price lower and upper bounds of each product are the minimum and maximum price of that product from the previous 30 days, respectively.

Second, following the company's policy, we can only change the price once a day for each product (instead of changing the price for every customer).

We collect the testing data from 01/02/2019 to 01/31/2019 (a total of 30 days). To better present the results, let $g \in \{0, 1\}$ denote the index of groups such that $g = 0$ represents the control group, and $g = 1$ represents the testing group. Then we calculate the average revenue $r_{g,t}$ per customer in day t for products in group g. The average revenue per customer is defined as the ratio between the collected revenue and the total number of unique visitors (including those who did not make a purchase) for group g in day t. Due to the data privacy policy of Alibaba, we will not be able to present the raw data of $r_{g,t}$. Instead, we will compute the percentage change in average revenue per customer, $r_{g,t}$, compared with the average revenue per customer of group g during the previous month $\bar{r}_g$. More specifically, we define

$$\Delta r_{g,t} := \frac{r_{g,t} - \bar{r}_g}{\bar{r}_g}, \qquad g = 0, 1.$$

To take away possible seasonal effects, our comparison will be between $\Delta r_{1,t}$ and $\Delta r_{0,t}$. The results are presented in Figure 2.8.

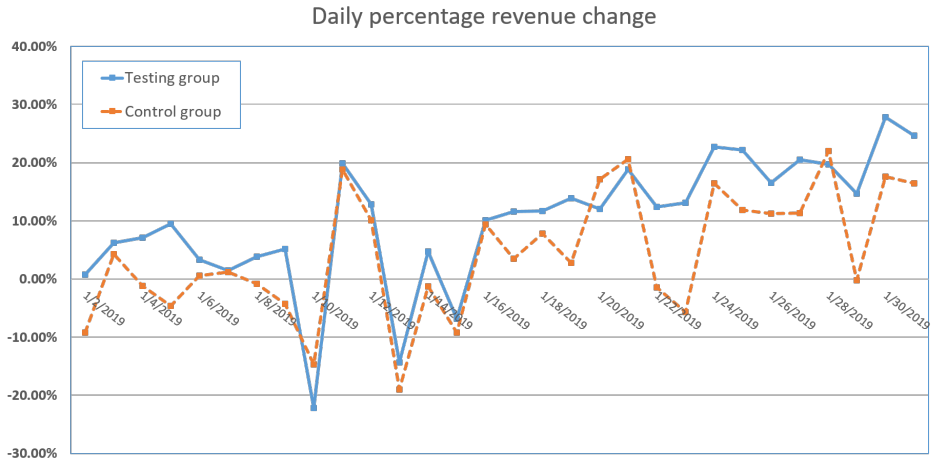


Figure 2.8: Comparison of $\Delta r_{g,t}$ between groups $g = 0, 1$ every day

As noted in the field experiment results in Figure 2.8, the percentage of increase of the average revenue per customer in the testing group is higher than that of the control group in 26 of the 30 days tested. By calculating the overall average revenue per customer for each group, we find that the average revenue per customer for the testing group is increased by 10.14% compared with the previous month, while in the control group, the average revenue per customer is increased by 4.39% compared with the previous month. Data further shows that our pricing policy helps to achieve this revenue increase by attracting more demand. Specifically, during the period of testing time, the purchasing probability

	01/02/19-01/31/19	
	Revenue	Demand
Testing group	10.14%	14.85%
Control group	4.39%	-0.05%

Table 2.3: Overall performance of two groups in the testing period. "Revenue" represents percentage change of average revenue, and "Demand" represents percentage change of purchasing probability.

of each customer is increased by 14.85% for the testing group, compared with -0.05% increase for the control group (see Table 2.3 for the summary). These results illustrate the effectiveness of our CSMP policy in boosting the revenue as compared with the current pricing policy of Alibaba.

2.5.4 Summary of numerical experiments

In this section we first present the simulation results using synthetic data under various scenarios to test the effectiveness and robustness of our algorithms, then we present the simulation results with real data from Alibaba using a more sophisticated ground truth demand model (for a more realistic simulation and robustness test under model misspecification). Finally we report the results from a field experiment conducted at Alibaba. The main findings from the numerical study are summarized as follows.

- In all the numerical results, pricing with clustering (either using our method in CSMP or classical K-means clustering with appropriate choice of K) outperforms the benchmarks of applying single-product pricing algorithm on each product or naively putting all products into a single cluster.

- Dynamic pricing with K-means clustering method sometimes works as effectively as (and at times even better than) our algorithm CSMP/CSMP-L. But its performance depends on the choice of the number of clusters K, which is unknown to the decision maker.

- The CSMP algorithm is quite robust under different scenarios: slightly different demand parameters within the same cluster, near static or slowly changing features, and misspecified ground truth demand model.

- The CSMP algorithm (with logistic demand function) showed satisfactory performance in the field experiment at Tmall Supermarket. Compared with products in the control group that used the business-as-usual pricing policy of Alibaba, the CSMP

algorithm significantly boosted the revenue of the testing products, demonstrating the effectiveness of the algorithm.

2.6 Proofs of Technical Results

In this section, we present all the missing proofs earlier in this chapter. We also prove the result discussed in Remark 2.3.4 of Section 2.3 for a more general definition of clusters.

2.6.1 Proof of Theorem 2.3.1

First of all, we define $\tilde{q}_j := \sum_{i \in \mathcal{N}_j} q_i$ as the probability that a customer views a product from cluster j. Then, define the events

$$\begin{aligned}
\mathcal{E}_{N,t} :=& \{\hat{\mathcal{N}}_t = \mathcal{N}_{i_t}\}, \\
\mathcal{E}_{B_j,t} :=& \{||\tilde{\theta}_{j,t} - \theta_j||_2 \leq \tilde{B}_{j,t}\}, \\
\mathcal{E}_{V,t} :=& \left\{ \lambda_{\min} \left(\sum_{s \in \tilde{\mathcal{T}}_{j_t,t}} u_s u_s' \right) \geq \frac{\lambda_1 \Delta_0^2 \sqrt{\tilde{q}_{j_t} t}}{8} \right\},
\end{aligned}$$

where $\lambda_1 = \min(1, \lambda_0)/(1 + \overline{p}^2)$ and $\tilde{\theta}_{j,t}$ is the estimated parameters using data from $\tilde{\mathcal{T}}_{j,t}$, and

$$\tilde{B}_{j,t} =: \frac{\sqrt{c(d+2)\log(1+t))}}{\sqrt{\lambda_{\min}(\tilde{V}_{j,t})}}$$

for some constant $c \geq 20/l_1^2$ and $\tilde{V}_{j,t} = I + \sum_{s \in \tilde{\mathcal{T}}_{j,t}} u_s u_s'$. These events hold at least with the following probabilities

$$\begin{aligned}
\mathbb{P}(\mathcal{E}_{N,t}) \geq& 1 - \frac{2n}{t^2} \qquad \text{for } t > \bar{t}, \\
\mathbb{P}(\mathcal{E}_{B_j,t}) \geq& 1 - \frac{1}{t} \qquad \text{for any } j \in [m], t \in \mathcal{T}, \\
\mathbb{P}(\mathcal{E}_{V,t}) \geq& 1 - \frac{7n}{t} \qquad \text{for } t > 2\bar{t},
\end{aligned}$$

where $\bar{t}$ is defined in (2.32). The first inequality is from our analysis after Lemma 2.6.5; the second inequality is from Corollary 2.6.1; the third inequality is from Lemma 2.6.6. We further define $\mathcal{E}_{B,t} = \bigcup_{j \in [m]} \mathcal{E}_{B_j,t}$, then it holds with probability at least $1 - m/t$ for any $t \in \mathcal{T}$. Now we define the event $\mathcal{E}_t$ as the union of $\mathcal{E}_{N,t}$, $\mathcal{E}_{B,t}$, and $\mathcal{E}_{V,t}$. This event holds with probability at least $1 - 10n/t$ obviously according to the probability of each event.

We split the regret by considering $t \leq 2\bar{t}$ and $t > 2\bar{t}$, i.e.,

$$\sum_{t=1}^{T} \mathbb{E}[r_t(p_t^*) - r_t(p_t)] = \sum_{t \leq 2\bar{t}} \mathbb{E}[r_t(p_t^*) - r_t(p_t)] + \sum_{t > 2\bar{t}} \mathbb{E}[r_t(p_t^*) - r_t(p_t)].$$

Obviously, the regret of the first summation can be bounded above by $2\overline{p}\bar{t}$. We focus on the second summation. For arbitrary $t > 2\bar{t}$,

$$\begin{aligned}
\mathbb{E}[r_t(p_t^*) - r_t(p_t)] =& \mathbb{E}[(r_t(p_t^*) - r_t(p_t))\mathbf{1}(\mathcal{E}_t)] + \mathbb{E}[(r_t(p_t^*) - r_t(p_t))\mathbf{1}(\bar{\mathcal{E}}_t)] \\
\leq& \mathbb{E}[(p_t^* \mu(\alpha_{i_t}' x_t + \beta_{i_t} p_t^*) - p_t \mu(\alpha_{i_t}' x_t + \beta_{i_t} p_t))\mathbf{1}(\mathcal{E}_t)] + \frac{10\overline{p}n}{t} \\
=& \mathbb{E}[(|2\beta_{i_t}\dot{\mu}(\alpha_{i_t}' x_t + \beta_{i_t}\bar{p}_t) + \beta_{i_t}^2 \bar{p}_t \ddot{\mu}(\alpha_{i_t}' x_t + \beta_{i_t}\bar{p}_t)|(p_t^* - p_t)^2)\mathbf{1}(\mathcal{E}_t)] \\
&+ \frac{10\overline{p}n}{t} \\
\leq& \mathbb{E}[(\tilde{L}_2(p_t^* - \tilde{p}_t - \Delta_t)^2)\mathbf{1}(\mathcal{E}_t)] + \frac{10\overline{p}n}{t} \\
\leq& 2\tilde{L}_2 L_0^2 \mathbb{E}[||\tilde{\theta}_{\mathcal{N}_t, t-1} - \theta_{i_t}||_2^2 \mathbf{1}(\mathcal{E}_t)] + 4\tilde{L}_2 \mathbb{E}[\Delta_t^2 \mathbf{1}(\mathcal{E}_t)] + \frac{10\overline{p}n}{t} \\
=& 2\tilde{L}_2 L_0^2 \mathbb{E}[||\tilde{\theta}_{j_t, t-1} - \theta_{j_t}||_2^2 \mathbf{1}(\mathcal{E}_t)] + 4\tilde{L}_2 \mathbb{E}[\Delta_t^2 \mathbf{1}(\mathcal{E}_t)] + \frac{10\overline{p}n}{t} \\
\leq& 2\tilde{L}_2 L_0^2 \mathbb{E}[\tilde{B}_{j_t, t-1}^2 \mathbf{1}(\mathcal{E}_t)] + 4\tilde{L}_2 \mathbb{E}[\Delta_t^2 \mathbf{1}(\mathcal{E}_t)] + \frac{10\overline{p}n}{t},
\end{aligned}$$

where the first inequality is from the probability of $\bar{\mathcal{E}}_t$, the second equality is by applying Taylor's theorem (where $\bar{p}_t$ is some price between p_t^* and p_t) with Assumption A-1 and Assumption A-2, the second inequality is from Assumption A-2 and $\tilde{L}_2$ is some constant depending on $L, L_1, L_2, \overline{p}$, and both the last equality and the last inequality are from the definition of $\mathcal{E}_t$ (i.e., events $\mathcal{E}_{N,t}$ and $\mathcal{E}_{B,t}$). Therefore, we have

$$\mathbb{E}[r_t(p_t^*) - r_t(p_t)] \leq 2\tilde{L}_2 L_0^2 \mathbb{E}[\tilde{B}_{j_t, t-1}^2 \mathbf{1}(\mathcal{E}_t)] + 4\tilde{L}_2 \mathbb{E}[\Delta_t^2 \mathbf{1}(\mathcal{E}_t)] + \frac{10\overline{p}n}{t}. \tag{2.20}$$

Summing over t, the sum of the last terms above obviously lead to the regret $O(n \log T)$. For the rest, we have

$$\begin{aligned}
\sum_{t > 2\bar{t}} \mathbb{E}[\tilde{B}_{j_t, t-1}^2 \mathbf{1}(\mathcal{E}_t)] \leq \frac{k_2 d \log T}{\Delta_0^2} \sum_{t > 2\bar{t}} \mathbb{E}\left[\frac{1}{\sqrt{\tilde{q}_{j_t} t}}\right] &= \frac{k_2 d \log T}{\Delta_0^2} \sum_{t > 2\bar{t}} \sum_{j \in [m]} \sqrt{\frac{\tilde{q}_j}{t}} \\
&\leq \frac{k_2 d \log T}{\Delta_0^2} \sum_{j \in [m]} \sqrt{\tilde{q}_j T} \leq \frac{k_2 d \log T}{\Delta_0^2} \sqrt{mT}
\end{aligned}$$

for some constant k_2, where the first inequality is from $\mathcal{E}_t$ (i.e., $\mathcal{E}_{V,t}$) and the definition of

$\tilde{B}^2_{j_t,t}$, the equality is by conditioning on $j_t = j$ for all $j \in [m]$, and the last inequality is because $\sum_j \tilde{q}_j = 1$ and apply Cauchy-Schwarz. Hence

$$\sum_{t>2\bar{t}} \mathbb{E}[\tilde{B}^2_{j_t,t-1}\mathbf{1}(\mathcal{E}_t)] \leq \frac{k_2 d \log T}{\Delta_0^2}\sqrt{mT}. \tag{2.21}$$

On the other hand, because $\hat{\mathcal{N}}_t = \mathcal{N}_{i_t}$ for all $t > 2\bar{t}$ on $\mathcal{E}_t$,

$$\sum_{t>2\bar{t}} \mathbb{E}[\Delta_t^2\mathbf{1}(\mathcal{E}_t)] \leq \sum_{j\in[m]} \mathbb{E}\left[\sum_{t\in\tilde{\mathcal{T}}_{j,T}} \frac{\Delta_0^2}{\sqrt{\tilde{T}_{j,t}}}\right] \leq \Delta_0^2 \sum_{j\in[m]} \mathbb{E}\left[\sqrt{\tilde{T}_{j,T}}\right] \leq \Delta_0^2\sqrt{mT}. \tag{2.22}$$

Putting (2.20), (2.21), and (2.22) together, we have

$$\sum_{t>2\bar{t}} \mathbb{E}[(r_t(p_t^*) - r_t(p_t))] \leq c_5 d \log(T)\sqrt{mT} + c_5 n \log T$$

for some constant c_5, and together with the regret for $t < 2\bar{t}$, we are done with the regret upper bound.

In the rest of this subsection, we prove the lemmas used in the proof of Theorem 2.3.1.

Lemma 2.6.1 *For each $j \in [m]$ and $t \in \mathcal{T}$, with probability at least $1 - \Delta$, $\tilde{T}_{j,t} \in [\tilde{q}_j t - \tilde{D}(t), \tilde{q}_j t + \tilde{D}(t)]$ for all $j \in [m]$, $t \in \mathcal{T}$, where $\tilde{D}(t) = \sqrt{t\log(2/\Delta)}$.*

Proof: Obviously $\tilde{T}_{j,t}$ is a binomial random variable with parameter t and $\tilde{q}_j$. Then we simply use Hoeffding inequality applied on sequence of i.i.d. Bernoulli random variable and a simple union bound on all $j \in [m]$ and $t \in \mathcal{T}$. □

Lemma 2.6.2 *For any $i \in [n]$ and $t \in \mathcal{T}$, let $V_{i,t} = I + \sum_{s\in\mathcal{T}_{i,t}} u_s u_s'$, we have that*

$$||\hat{\theta}_{i,t} - \theta_i||_{V_{i,t}} \leq \frac{2\sqrt{(d+2)\log(1 + T_{i,t}R^2/(d+2)) + 2\log(1/\Delta)} + 2l_1 L}{l_1}$$

with probability at least $1 - \Delta$.

Proof: We first fix some $i \in [n]$, and we drop the index dependency on i for convenience of notation. At round s, the gradient of likelihood function $\nabla l_s(\phi)$ is equal to

$$\nabla l_s(\phi) = (\mu(u_s'\phi) - d_s)u_s. \tag{2.23}$$

And its Hessian is

$$\nabla^2 l_s(\phi) = \dot{\mu}(u_s'\phi)u_s u_s'. \tag{2.24}$$

Applying Taylor's theorem, we obtain

$$\begin{aligned} 0 &\geq \sum_s l_s(\hat{\theta}_t) - l_s(\theta) \\ &= \sum_s \nabla l_s(\theta)'(\hat{\theta}_t - \theta) + \frac{1}{2}\sum_s \dot{\mu}(u_s'\bar{\theta}_t)(u_s'(\hat{\theta}_t - \theta))^2 + \frac{l_1}{2}||\hat{\theta}_t - \theta||_2^2 - \frac{l_1}{2}||\hat{\theta}_t - \theta||_2^2, \end{aligned} \tag{2.25}$$

where the first inequality is from the optimality of $\hat{\theta}_t$, and $\bar{\theta}_t$ is a point on line segment between $\hat{\theta}_t$ and θ. Note that by our assumption and boundedness of u_s and θ, we have $\dot{\mu}(u_s'\bar{\theta}_t) \geq l_1$. Therefore, we have

$$\sum_s \dot{\mu}(u_s'\bar{\theta}_t)(u_s'(\hat{\theta}_t - \theta))^2 + l_1||\hat{\theta}_t - \theta||_2^2 \geq l_1||\hat{\theta}_t - \theta||_{V_t}^2, \tag{2.26}$$

where $V_t = I + \sum_s u_s u_s'$. On the other hand, we have

$$\nabla l_s(\theta_i) = -\epsilon_s u_s, \tag{2.27}$$

where ϵ_s is the zero-mean error, which is obviously sub-Gaussian with parameter 1 as it is bounded.

Now combining (2.25), (2.26), and (2.27), we have

$$\frac{l_1}{2}||\hat{\theta}_t - \theta||_{V_t}^2 \leq \sum_s \epsilon_s u_s'(\hat{\theta}_t - \theta) + 2l_1 L^2 \leq ||\hat{\theta}_t - \theta||_{V_t}||Z_t||_{V_t^{-1}} + 2l_1 L^2, \tag{2.28}$$

where $Z_t := \sum_s \epsilon_s u_s$, and the second inequality is from Cauchy-Schwarz and $||\hat{\theta}_t - \theta||_2 \leq 2L$. This leads to $||\hat{\theta}_t - \theta||_{V_t} \leq \frac{2}{l_1}||Z_t||_{V_t^{-1}} + 2L$.

To bound $||Z_t||_{V_t^{-1}}$, according to Theorem 1 in Abbasi-Yadkori et al. (2011), we have

$$||Z_t||_{V_t^{-1}} \leq \sqrt{(d+2)\log(1 + \frac{T_{i,t}R^2}{d+2}) + 2\log(1/\Delta)}$$

with probability at least $1 - \Delta$ and we are done. □

Corollary 2.6.1 *For any $j \in [m]$ and $t \in \mathcal{T}$, let $\tilde{V}_{j,t} := I + \sum_{s \in \tilde{\mathcal{T}}_{j,t}} u_s u_s'$, we have that*

$$||\tilde{\theta}_{j,t} - \theta_j||_{\tilde{V}_{j,t}} \leq \frac{2\sqrt{(d+2)\log(1+\tilde{T}_{j,t}R^2/(d+2)) + 2\log(1/\Delta)} + 2l_1 L}{l_1}$$

with probability at least $1 - \Delta$.

Next result is the minimum eigenvalue of the Fisher's information matrix.

Lemma 2.6.3 *Let $u_t' = (x_t', \tilde{p}_t + \Delta_t)$ where Δ_t is a zero mean error with variance satisfying $E[\Delta_t^2 | \mathcal{F}_{t-1}] = \omega_t > 0$, we must have $\lambda_{\min}(\mathbb{E}[u_t u_t' | \mathcal{F}_{t-1}]) \geq \omega_t \min[1, \lambda_0]/(1+\overline{p}^2) > 0$. So we can set $\lambda_{\min}(\mathbb{E}[u_t u_t' | \mathcal{F}_{t-1}]) \geq \lambda_1 \omega_t$ for some constant $\lambda_1 = \min[1, \lambda_0]/(1+\overline{p}^2)$.*

Proof: Note that the Fisher's information matrix can be written as

$$\mathbb{E}[u_t u_t' | \mathcal{F}_{t-1}] = \begin{bmatrix} 1 & 0 & \tilde{p}_t \\ 0 & \Sigma_z & 0 \\ \tilde{p}_t & 0 & \tilde{p}_t^2 + \mu_t \end{bmatrix}$$

which is a submatrix of the matrix

$$M := \begin{bmatrix} 1 & 0 & \tilde{p}_t & 0 \\ 0 & \Sigma_z & 0 & \tilde{p}_t \Sigma_z \\ \tilde{p}_t & 0 & \tilde{p}_t^2 + \omega_t & 0 \\ 0 & \tilde{p}_t \Sigma_z & 0 & (\tilde{p}_t^2 + \omega_t)\Sigma_z \end{bmatrix} = M_p \otimes M_z$$

where

$$M_p = \begin{bmatrix} 1 & \tilde{p}_t \\ \tilde{p}_t & \tilde{p}_t^2 + \omega_t \end{bmatrix}, \qquad M_z = \begin{bmatrix} 1 & 0 \\ 0 & \Sigma_z \end{bmatrix},$$

and $\otimes$ is the Kronecker product.

To derive the minimum eigenvalue of M_p, note that it is just a 2×2 matrix so we can easily compute that

$$\lambda_{\min}(M_p) = \frac{(\tilde{p}_t^2 + \omega_t + 1)(1 - \sqrt{1 - 4\omega_t/(\tilde{p}_t^2 + \omega_t + 1)^2})}{2} \geq \frac{\omega_t}{\tilde{p}_t^2 + \omega_t + 1} \geq \frac{\omega_t}{1 + \overline{p}^2}.$$

For M_z, let $y' = (y_1, y_2') \in \mathbb{R}^{d+1}$ where $y_1 \in \mathbb{R}$ and $y_2 \in \mathbb{R}^d$, then

$$y' M_z y = y_1^2 + y_2' \Sigma_z y_2 \geq y_1^2 + \lambda_0 ||y_2||_2^2 \geq \min[1, \lambda_0] \, ||y||_2^2.$$

Therefore, $\lambda_{\min}(M_z) \geq \min[1, \lambda_0] > 0$.

According to Theorem 4.2.12 in Horn et al. (1990), we have

$$\lambda_{\min}(M) = \lambda_{\min}(M_p)\lambda_{\min}(M_z) \geq \frac{\omega_t}{1+\bar{p}^2} \min[1, \lambda_0].$$

Then we obtain the result as $\mathbb{E}[u_t u_t']$ is the submatrix of M. □

We apply a matrix concentration inequality result and obtain the minimum eigenvalue of the empirical Fisher's information matrix.

Lemma 2.6.4 *For any $i \in [n]$ and*

$$t > \left(\frac{8R\log((d+2)T)}{\lambda_1 \Delta_0^2 \min_{i\in[n]} q_i} \right)^2,$$

where $R := 2 + \bar{p}^2$, we have

$$\mathbb{P}\left(\lambda_{\min}\Big(\sum_{s\in\mathcal{T}_{i,t}} u_s u_s' \Big) < \frac{\lambda_1 \Delta_0^2 q_i \sqrt{t}}{2} \right) < \frac{1}{t^2}.$$

Proof: Note that $\lambda_{\max}(u_s u_s') = ||u_s||_2^2 \leq R = 2 + \bar{p}^2$. We find that

$$\sum_{s\in\mathcal{T}_{i,t}} u_s u_s' = \sum_{s=1}^{t} \mathbb{1}(i_s = i) u_s u_s',$$

and, by Lemma 2.6.3,

$$\lambda_{\min}(\mathbb{E}[\mathbb{1}(i_s = i) u_s u_s' | \mathcal{F}_{s-1}]) = q_i \lambda_{\min}(\mathbb{E}[u_s u_s' | \mathcal{F}_{s-1}]) \geq \lambda_1 q_i \omega_s.$$

Therefore,

$$\begin{aligned}\lambda_{\min}\left(\sum_{s=1}^{t} \mathbb{E}[\mathbb{1}(i_s = i) u_s u_s' | \mathcal{F}_{s-1}] \right) \geq & \sum_{s=1}^{t} \lambda_{\min}\left(\mathbb{E}[\mathbb{1}(i_s = i) u_s u_s' | \mathcal{F}_{s-1}] \right) \\ \geq & q_i \lambda_1 \sum_{s=1}^{t} \omega_s \geq q_i \lambda_1 \Delta_0^2 \frac{t}{\sqrt{t}} \geq q_i \lambda_1 \Delta_0^2 \sqrt{t}.\end{aligned}$$

As a result, we have that

$$
\begin{aligned}
&\mathbb{P}\left(\lambda_{\min}(\sum_{s\in\mathcal{T}_{i,t}} u_s u_s') < \frac{\lambda_1\Delta_0^2 q_i\sqrt{t}}{2}\right)\\
=&\mathbb{P}\left(\lambda_{\min}(\sum_{s\in\mathcal{T}_{i,t}} u_s u_s') < \frac{\lambda_1\Delta_0^2 q_i\sqrt{t}}{2}, \sum_{s=1}^{t}\lambda_{\min}\left(\mathbb{E}[\mathbb{1}(i_s=i)u_s u_s'|\mathcal{F}_{s-1}]\right) \geq \lambda_1\Delta_0^2 q_i\sqrt{t}\right)\\
\leq&\mathbb{P}\left(\lambda_{\min}(\sum_{s\in\mathcal{T}_{i,t}} u_s u_s') < \frac{\lambda_1\Delta_0^2 q_i\sqrt{t}}{2}, \lambda_{\min}\left(\sum_{s=1}^{t}\mathbb{E}[\mathbb{1}(i_s=i)u_s u_s'|\mathcal{F}_{s-1}]\right) \geq \lambda_1\Delta_0^2 q_i\sqrt{t}\right)\\
\leq&(d+2)e^{-\frac{\lambda_1\Delta_0^2 q_i\sqrt{t}}{4R}},
\end{aligned}
$$

where the last inequality is from Theorem 3.1 in Tropp (2011) with $\zeta = 1/2$.

So for any $i \in [n]$ and

$$t > \left(\frac{8R\log(T(d+2))}{\lambda_1\Delta_0^2\min_{i\in[n]} q_i}\right)^2,$$

we have the simple union bound over $i \in [n], t \in \mathcal{T}$, $(d+2)\exp(-\lambda_1\Delta_0^2 q_i\sqrt{t}/(4R)) < 1/t^2$, and the proof is complete. □

Clearly, if we combine Lemma 2.6.4 and Lemma 2.6.2, for any $i \in [n]$, $t > \bar{t}_1$ where

$$\bar{t}_1 = \left(\frac{8R\log(T(d+2))}{\lambda_1\Delta_0^2\min_{i\in[n]} q_i}\right)^2, \tag{2.29}$$

we have that

$$
\begin{aligned}
||\hat{\theta}_{i,t} - \theta_i||_2 &\leq \frac{2\sqrt{(d+2)\log(1+tR^2/(d+2)) + 2\log t^2} + 2l_1 L}{l_1\sqrt{\lambda_{\min}(V_{i,t})}} \\
&\leq \frac{\sqrt{c(d+2)\log(1+t)}}{\sqrt{\lambda_{\min}(V_{i,t})}} = B_{i,t}
\end{aligned} \tag{2.30}
$$

for some constant $c > 20/l_1^2$, and

$$B_{i,t} \leq \frac{\sqrt{2c(d+2)\log(1+t)}}{\Delta_0\sqrt{\lambda_1 q_i\sqrt{t}}} \tag{2.31}$$

with probability at least $1 - 2/t^2$.

The next lemma states that when estimation errors are bounded, under certain conditions we have $\hat{\mathcal{N}}_t = \mathcal{N}_{i_t}$.

Lemma 2.6.5 *Suppose for all $i \in [n]$ it holds that $||\hat{\theta}_{i,t-1}-\theta_i||_2 \leq B_{i,t-1}$ and $B_{i,t-1} < \gamma/4$. Then*

$$\hat{\mathcal{N}}_t = \mathcal{N}_{i_t}.$$

Proof: First of all, for $i_1, i_2 \in [n]$, if they belong to different clusters and $B_{i_1,t-1} + B_{i_2,t-1} < \gamma/2$, we must have $||\hat{\theta}_{i_1,t-1} - \hat{\theta}_{i_2,t-1}||_2 > B_{i_1,t-1} + B_{i_2,t-1}$ because

$$\begin{aligned}\gamma \leq & ||\theta_{i_1} - \theta_{i_2}||_2 \leq ||\theta_{i_1} - \hat{\theta}_{i_1,t-1}||_2 + ||\hat{\theta}_{i_1,t-1} - \hat{\theta}_{i_2,t-1}||_2 + ||\hat{\theta}_{i_2,t-1} - \theta_{i_2}||_2 \\ \leq & B_{i_1,t-1} + ||\hat{\theta}_{i_1,t-1} - \hat{\theta}_{i_2,t-1}||_2 + B_{i_2,t-1} < \gamma/2 + ||\hat{\theta}_{i_1,t-1} - \hat{\theta}_{i_2,t-1}||_2,\end{aligned}$$

which implies that $||\hat{\theta}_{i_1,t-1} - \hat{\theta}_{i_2,t-1}||_2 > \gamma/2 > B_{i_1,t-1} + B_{i_2,t-1}$.

On the other hand, if $||\hat{\theta}_{i_1,t-1}-\hat{\theta}_{i_2,t-1}||_2 > B_{i_1,t-1}+B_{i_2,t-1}$, we must have i_1, i_2 belongs to different clusters because

$$\begin{aligned}B_{i_1,t-1} + B_{i_2,t-1} < & ||\hat{\theta}_{i_1,t-1} - \hat{\theta}_{i_2,t-1}||_2 \\ \leq & ||\theta_{i_1} - \hat{\theta}_{i_1,t-1}||_2 + ||\hat{\theta}_{i_1,t-1} - \hat{\theta}_{i_2,t-1}||_2 + ||\hat{\theta}_{i_2,t-1} - \theta_{i_2}||_2 \\ \leq & B_{i_1,t-1} + ||\hat{\theta}_{i_1,t-1} - \hat{\theta}_{i_2,t-1}||_2 + B_{i_2,t-1},\end{aligned}$$

which implies $||\hat{\theta}_{i_1,t-1} - \hat{\theta}_{i_2,t-1}||_2 > 0$, i.e., they belong to different clusters.

Therefore, if $i \in \hat{\mathcal{N}}_t$, i.e., $||\hat{\theta}_{i_t,t-1} - \hat{\theta}_{i,t-1}|| \leq B_{i_t,t-1} + B_{i,t-1}$, we must have that $i \in \mathcal{N}_{i_t}$ as well or $B_{i_t,t-1} + B_{i,t-1} \geq \gamma/2$ (which is impossible by our assumption that $B_{i,t-1} < \gamma/4$).

On the other hand, if $i \in \mathcal{N}_{i_t}$, then we must have $||\hat{\theta}_{i_t,t-1} - \hat{\theta}_{i,t-1}|| \leq B_{i_t,t-1} + B_{i,t-1}$, which implies that $i \in \hat{\mathcal{N}}_t$ as well.

Above all, we have shown that $\hat{\mathcal{N}}_{i_t} = \mathcal{N}_{i_t}$. □

Note that given (2.30) and (2.31), we have that $B_{i,t-1} < \gamma/4$ for all i if

$$t > 1 + \frac{k_1((d+2)\log(1+T))^2}{\gamma^4\lambda_1^2\Delta_0^4 \min_{i\in[n]} q_i^2}$$

for some constant k_1. Therefore, for each $t > \bar{t}$ where

$$\bar{t} = \max\left\{4\bar{t}_1, 1 + \frac{k_1((d+2)\log(1+T))^2}{\gamma^4\lambda_1^2\Delta_0^4 \min_{i\in[n]} q_i^2}\right\}, \tag{2.32}$$

and $\bar{t}_1$ is defined in (2.29), $\hat{\mathcal{N}}_t = \mathcal{N}_{i_t}$ with probability at least $1 - 2n/t^2$.

The next lemma shows that the clustered estimation will be quite accurate when most of the $\hat{\mathcal{N}}_t$ is actually equal to $\mathcal{N}_{i_t}$.

Lemma 2.6.6 *For any t such that $t > 2\bar{t}$, we have*

$$\mathbb{P}\left(\lambda_{\min}\left(\sum_{s\in\tilde{\mathcal{T}}_{j_t,t}} u_s u_s'\right) < \frac{\lambda_1\Delta_0^2\sqrt{\tilde{q}_{j_t}t}}{8}\right) < \frac{7n}{t},$$

where $\bar{t}$ is defined in (2.32).

Proof: The proof is analogous to Lemma 2.6.4. Let $\mathcal{E}_{N,t}$ be the event such that $\hat{\mathcal{N}}_t = \mathcal{N}_{i_t}$, and $\tilde{\mathcal{E}}_{j,t}$ be the event such that $\tilde{T}_{j,t} \le 3\tilde{q}_j t/2$. From our previous analysis, we know that given $t > \bar{t}$, $\mathcal{E}_{N,t}$ holds with probability at least $1-2n/t^2$. Also, according to Lemma 2.6.1, event $\tilde{\mathcal{E}}_{j,t}$ holds with probability at least $1-1/t^2$ given $t \ge 8\log(2T)/\min_{j\in[m]}\tilde{q}_j^2$ (which is satisfied by taking $t > \bar{t}$).

On event $\tilde{\mathcal{E}}_{j,t}$ and $\mathcal{E}_{N,s}$ for all $s \in [t/2, t]$ (which holds with probability at least $1-6n/t$), we have

$$\lambda_{\min}(\mathbb{E}[\mathbb{1}(j_s = j)u_s u_s'|\mathcal{F}_{s-1}]) \ge \lambda_1\tilde{q}_j\omega_s = \lambda_1\Delta_0^2\tilde{q}_j(\tilde{T}_{j,s})^{-1/2} \ge \lambda_1\Delta_0^2\sqrt{\frac{2\tilde{q}_j}{3t}}$$

by Lemma 2.6.3 and definition of $\tilde{q}_j$. This implies that

$$\lambda_{\min}\left(\sum_{s=1}^{t}\mathbb{E}[\mathbb{1}(j_s = j)u_s u_s'|\mathcal{F}_{s-1}]\right) \ge \sum_{s=t/2}^{t}\lambda_{\min}\left(\mathbb{E}[\mathbb{1}(j_s = j)u_s u_s'|\mathcal{F}_{s-1}]\right) \ge \lambda_1\Delta_0^2\frac{\sqrt{\tilde{q}_j t}}{4}.$$

Therefore, we have for any $t > 2\bar{t}$,

$$\begin{aligned}
&\mathbb{P}\left(\lambda_{\min}\left(\sum_{s\in\tilde{\mathcal{T}}_{j_t,t}} u_s u_s'\right) < \frac{\lambda_1\Delta_0^2\sqrt{\tilde{q}_{j_t}t}}{8}\right)\\
&= \sum_{j\in[m]}\mathbb{P}\left(\lambda_{\min}\left(\sum_{s\in\tilde{\mathcal{T}}_{j_t,t}} u_s u_s'\right) < \frac{\lambda_1\Delta_0^2\sqrt{\tilde{q}_{j_t}t}}{8}\middle| j_t = j\right)\mathbb{P}(j_t = j)\\
&= \sum_{j\in[m]}\mathbb{P}\left(\lambda_{\min}\left(\sum_{s\in\tilde{\mathcal{T}}_{j,t}} u_s u_s'\right) < \frac{\lambda_1\Delta_0^2\sqrt{\tilde{q}_j t}}{8}\right)\tilde{q}_j.
\end{aligned}$$

For each $j \in [m]$, we have

$$\begin{aligned}
&\mathbb{P}\left(\lambda_{\min}\left(\sum_{s\in\tilde{\mathcal{T}}_{j,t}} u_s u_s'\right) < \frac{\lambda_1\Delta_0^2\sqrt{\tilde{q}_j t}}{8}\right)\\
\leq&\mathbb{P}\left(\lambda_{\min}\left(\sum_{s\in\tilde{\mathcal{T}}_{j,t}} u_s u_s'\right) < \frac{\lambda_1\Delta_0^2\sqrt{\tilde{q}_j t}}{8}, \bigcup_{s\in[t/2,t]}(\mathcal{E}_{N,t}\cup\tilde{\mathcal{E}}_{j,t})\right) + \frac{6n}{t}\\
=&\mathbb{P}\left(\lambda_{\min}\left(\sum_{s\in\tilde{\mathcal{T}}_{j,t}} u_s u_s'\right) < \frac{\lambda_1\Delta_0^2\sqrt{\tilde{q}_j t}}{8}, \lambda_{\min}\left(\sum_{s\in\tilde{\mathcal{T}}_{j,t}} \mathbb{E}[u_s u_s'|\mathcal{F}_{s-1}]\right) \geq \frac{\lambda_1\Delta_0^2\sqrt{\tilde{q}_j t}}{4},\right.\\
&\left.\bigcup_{s\in[t/2,t]}(\mathcal{E}_{N,t}\cup\tilde{\mathcal{E}}_{j,t})\right) + \frac{6n}{t} \leq \frac{7n}{t},
\end{aligned}$$

where the first inequality is from the probability of the complement of $\bigcup_{s\in[t/2,t]}(\mathcal{E}_{N,t}\cup\tilde{\mathcal{E}}_{j,t})$, and the last inequality is by Theorem 3.1 in Tropp (2011), and we take

$$t > \left(\frac{8R\log(2(d+2)T)}{\lambda_1\Delta_0^2\min_{j\in[m]}\sqrt{\tilde{q}_j}}\right)^2.$$

Since $\bar{t} > \left(8R\log(2(d+2)T)/(\lambda_1\Delta_0^2\min_{j\in[m]}\sqrt{\tilde{q}_j})\right)^2$ by definition, we complete the proof. □

2.6.2 Proofs for the linear model

Proof of Theorem 2.4.1. First of all, we define event

$$\begin{aligned}
\tilde{\mathcal{E}}_t := \Big\{&|\tilde{\beta}_{j_t,t} - \beta_{j_t}| \leq k_8\sqrt{\log t}(\tilde{q}_{j_t}t)^{-1/4}/\Delta_0,\\
&|\tilde{\alpha}_{j_t,t}'x - \alpha_{j_t}'x| \leq k_7\sqrt{(d+1)}\log t(\tilde{q}_{j_t}t)^{1/4}/\Delta_0||x||_{\tilde{V}_{j_t,t}^{-1}}\Big\}.
\end{aligned}$$

According to Lemma 2.6.11, this event holds with probability at least $1 - 7n/t$ for any $t > 2\bar{t}'$ where

$$\bar{t}' = O\left(\left(\frac{\sqrt{d}\log T}{\min_{i\in[n]} q_i^{\kappa/2}}\right)^{4/(2\kappa-1)}\right)$$

is defined in (2.36).

Therefore, we can split the regret into $t \leq 2\bar{t}'$ (which has regret at most $O(\bar{t}')$) and $t > 2\bar{t}'$. Note that for any $t > 2\bar{t}'$, on event $\tilde{\mathcal{E}}_t$ and $\mathcal{E}_{N,t}$ (such that $\hat{\mathcal{N}}_t = \mathcal{N}_{i_t}$, which holds

with probability at least $1 - 2n/t^2$ according to Lemma 2.6.10), we have

$$\begin{aligned} r_t(p_t^*) - r_t(p_t) \leq & -\beta_{i_t}(p_t^* - \tilde{p}_t - \Delta_t)^2 \leq -2\beta_{i_t}(|p_t^* - p_t'| + |\Delta_t|)^2 - 2\beta_{i_t}\Delta_t^2 \\ \leq & c_7((\alpha_{i_t}' x_t - \tilde{\alpha}_{j_t,t-1}' x_t)^2 + (\beta_{i_t} - \tilde{\beta}_{j_t,t-1})^2 + \Delta_t^2) \\ \leq & c_7(\tilde{C}_{j_t,t-1}^{\alpha}(x_t))^2 + c_8 \log T(\Delta_0^2(\tilde{T}_{j_t,t})^{-1/2} + (\tilde{q}_{j_t} t)^{-1/2}/\Delta_0^2) \end{aligned}$$

for some constants c_7, c_8, where the third inequality is from the definition of optimal price given demand parameters and covariates, and the fourth inequality is from Cauchy-Schwarz, event $\tilde{\mathcal{E}}_t$, and the definition of Δ_t. Here $\tilde{C}_{j_t,t-1}^{\alpha}(x_t)$ is defined as

$$\tilde{C}_{j_t,t-1}^{\alpha}(x_t) := k_7\sqrt{(d+1)}\log(t-1)(\tilde{q}_{j_t}(t-1))^{1/4}/\Delta_0||x||_{\tilde{V}_{j_t,t-1}^{-1}}.$$

For the second terms, if we sum them up over t, their summation can be bounded by $c_9 \log T\sqrt{mT}$ for some constant c_9 as we did in the proof of Theorem 2.3.1. For the first term, there is some constant c_{10} such that

$$(\tilde{C}_{j_t,t-1}^{\alpha}(x_t))^2 \leq c_{10}(d+1)\log^2 T\sqrt{\tilde{q}_{j_t} t}||x||_{\tilde{V}_{j_t,t}^{-1}}^2. \tag{2.33}$$

If we sum them over t, we have (on events $\tilde{\mathcal{E}}_t$ and $\mathcal{E}_{N,t}$)

$$\begin{aligned} \sum_{t>2\bar{t}'} \mathbb{E}\left[\sqrt{\tilde{q}_{j_t} t}||x_t||_{\tilde{V}_{j_t,t}^{-1}}^2\right] \leq & \sum_{j\in[m]} \sqrt{\tilde{q}_j T}\mathbb{E}\left[\sum_{t>2\bar{t}', t\in\tilde{T}_{j,T}} ||x_t||_{\tilde{V}_{j,t}^{-1}}^2\right] \\ \leq & c_{11}(d+1)\log T \sum_{j\in[m]} \sqrt{\tilde{q}_j T} \leq c_{11}(d+1)\log T\sqrt{mT} \end{aligned}$$

for some constant c_{11} where the second inequality is by Lemma 11 in Abbasi-Yadkori et al. (2011). Therefore, combined with (2.33), its summation over $t > 2\bar{t}'$ is at most $O\left(d^2 \log^3 T\sqrt{mT}\right)$. Note that since the expected regret incurred on any of events $\tilde{\mathcal{E}}_t$ or $\mathcal{E}_{N,t}$ fail is at most $O(n \log T)$, we finish the proof.

In the rest of this subsection, we prove several lemmas that are needed for the proof of Theorem 2.4.1. The first lemma is about length of $T_{i,t}$.

Lemma 2.6.7 *For any $i \in [n], t \in \mathcal{T}$, with probability at least $1 - \Delta$, $T_{i,t} \in [q_i t - D(t), q_i t + D(t)]$, where $D(t) = \sqrt{t\log(2/\Delta)/2}$.*

Proof: Proof is the same as Lemma 2.6.1 hence neglected. □

Lemma 2.6.8 *For any $\mathcal{T}_{t_1,t_2} := \{t_1+1,\dots,t_2\}$ and $j \in [m]$, we have*

$$|\tilde{\mathcal{T}}_{j,t_2} \cap \mathcal{T}_{t_1,t_2}| \in [\tilde{q}_j(t_2-t_1) + \tilde{D}(t_2-t_1), \tilde{q}_j(t_2-t_1) - \tilde{D}(t_2-t_1)]$$

with probability at least $1-\Delta$ where $\tilde{D}(t) = \sqrt{t\log(2/\Delta)}$.

Proof: This is an immediate result of Lemma 2.6.1 and Lemma 2.6.7. □

Lemma 2.6.9 *For any $i \in [n], t \in \mathcal{T}$, we have that*

$$\begin{aligned}
|\hat{\beta}_{i,t} - \beta_i| \leq & k_4\sqrt{\log(1/\Delta)+\log(1+t)}\left(\sum_{s\in\mathcal{T}_{i,t}}\Delta_s^2\right)^{-1/2} \\
||\hat{\alpha}_{i,t} - \alpha_i||_{V_{i,t}} \leq & k_3\sqrt{d+1}(\log(1/\Delta)+\log(1+t))\left(\sum_{s\in\mathcal{T}_{i,t}}\Delta_s^2\right)^{-1/2}\sqrt{T_{i,t}}
\end{aligned}$$

for some constant k_3, k_4 with probability at least $1-\Delta$. In particular, we can show that $|\hat{\beta}_{i,t} - \beta_i| \leq \tilde{C}^{\beta}_{i,t}$ and $||\hat{\alpha}_{i,t} - \alpha_i||_{V_{i,t}} \leq \tilde{C}^{\alpha}_{i,t}$ with probability at least $1-1/t^2$.

Proof: First of all, we drop the index dependency on i for the sake of convenience. According to definition of $\hat{\beta}_t$, we have that

$$\hat{\beta}_t - \beta = \frac{\sum_{s\in\mathcal{T}_t} k_s\Delta_s}{\sum_{s\in\mathcal{T}_t}\Delta_s^2},$$

where $k_s := \alpha' x_s + \beta\tilde{p}_s + \epsilon_s$ which satisfies $|k_s| \leq \tilde{L} := 2L + \overline{p}L + 1$ by the boundedness assumption.

We can write $k_s\Delta_s = |\Delta_s|k_s\sigma_s$ where $\sigma_s = \pm 1$ with probability $1/2$, and

$$\begin{aligned}
|\hat{\beta}_t - \beta|\sqrt{\sum_{s\in\mathcal{T}_t}|\Delta_s|^2} = & \frac{|\sum_{s\in\mathcal{T}_t} k_s\sigma_s|\Delta_s||}{\sqrt{\Delta_0^2/\sqrt{t} + \sum_{s\in\mathcal{T}_t} k_s^2|\Delta_s|^2}} \frac{\sqrt{\Delta_0^2/\sqrt{t} + \sum_{s\in\mathcal{T}_t} k_s^2|\Delta_s|^2}}{\sqrt{\sum_{s\in\mathcal{T}_t}|\Delta_s|^2}} \\
\leq & \sqrt{1+\tilde{L}^2}\frac{|\sum_{s\in\mathcal{T}_t} k_s\sigma_s|\Delta_s||}{\sqrt{\Delta_0^2/\sqrt{t} + \sum_{s\in\mathcal{T}_t} k_s^2|\Delta_s|^2}},
\end{aligned}$$

where the inequality is because $|\Delta_s| \geq \Delta_0^2/\sqrt{t}$ for any $s \leq t$. Both σ_s and $k_s|\Delta_s|$ are adapted to filtration $\{\mathcal{F}_s\}$, and σ_s, which is sub-Gaussian with parameter 1, form a martingale difference sequence. Then Theorem 1 in Abbasi-Yadkori et al. (2011) (applied on

single dimensional case) gives us that

$$|\hat{\beta}_t - \beta|\sqrt{\sum_{s\in\mathcal{T}_t}\Delta_s^2} \leq \sqrt{1+\tilde{L}^2}\sqrt{\log(\sum_{s\in\mathcal{T}_t} k_s^2\Delta_s^2 + \Delta_0^2/\sqrt{t}) - \log(\Delta_0^2/\sqrt{t}) + 2\log(2/\Delta)}$$
$$\leq k_4\sqrt{\log(1/\Delta) + \log(1+t)} \tag{2.34}$$

with probability at least $1-\Delta/2$ for some constant k_4.

On the other hand, by definition of $\hat{\alpha}_t$,

$$(\sum_{s\in\mathcal{T}_t} x_s x_s' + I)(\hat{\alpha}_t - \alpha) = (\sum_{s\in\mathcal{T}_t} p_s x_s)(\beta - \hat{\beta}_t) + \sum_{s\in\mathcal{T}_t}\epsilon_s x_s + \alpha,$$

which implies that

$$\begin{aligned}||\hat{\alpha}_t - \alpha||_{V_t} \leq& ||\sum_{s\in\mathcal{T}_t} p_s x_s||_{V_t^{-1}}|\beta - \hat{\beta}_t| + ||\sum_{s\in\mathcal{T}_t}\epsilon_s x_s||_{V_t^{-1}} + L\\ \leq& \overline{p}\sum_{s\in\mathcal{T}_t}||x_s||_{V_t^{-1}}|\beta - \hat{\beta}_t| + ||\sum_{s\in\mathcal{T}_t}\epsilon_s x_s||_{V_t^{-1}} + L\\ \leq& \overline{p}|\beta - \hat{\beta}_t|\sqrt{2T_t(d+1)\log(1+2T_t/(d+1))}\\ &+ \sqrt{(d+1)\log(1+2T_t/(d+1)) + 2\log(2/\Delta)} + L,\end{aligned}$$

where $V_t = I + \sum_{s\in\mathcal{T}_t} x_s x_s'$, and the last inequality hold with probability at least $1-\Delta/2$ according to Theorem 1 and Lemma 11 in Abbasi-Yadkori et al. (2011). Then taking some appropriate k_3' gives us the bound, i.e.,

$$||\hat{\alpha}_t - \alpha||_{V_t} \leq k_3'(\sqrt{(d+1)(\log(1/\Delta) + \log(1+t))}|\beta - \hat{\beta}_t|\sqrt{T_t} + 1). \tag{2.35}$$

Therefore, events (2.34) and (2.35) hold together with probability at least $1-\Delta$.

According to the result above, we can take $\Delta = 1/t^2$ and let c_1, c_2 in (2.17) chosen appropriately such that $|\hat{\beta}_{i,t} - \beta_i| \leq \tilde{C}_{i,t}^{\beta}$ and $||\hat{\alpha}_{i,t} - \alpha_i||_{V_{i,t}} \leq \tilde{C}_{i,t}^{\alpha}$ with probability at least $1 - 1/t^2$. □

Corollary 2.6.2 *For any $j \in [m], t \in \mathcal{T}$, we have that*

$$|\tilde{\beta}_{j,t} - \beta_i| \leq k_4 \sqrt{\log(1/\Delta) + \log(1+t)} \left(\sum_{s \in \tilde{\mathcal{T}}_{j,t}} \Delta_s^2 \right)^{-1/2}$$

$$||\tilde{\alpha}_{j,t} - \alpha_i||_{\tilde{V}_{j,t}} \leq k_3 \sqrt{d+1} (\log(1/\Delta) + \log(1+t)) \left(\sum_{s \in \tilde{\mathcal{T}}_{j,t}} \Delta_s^2 \right)^{-1/2} \sqrt{\tilde{T}_{j,t}}$$

with probability at least $1 - \Delta$.

Lemma 2.6.10 *For any t such that*

$$t > \bar{t}' := \max \left\{ \frac{2t_0}{\min_i q_i}, \frac{4\log(2T)}{\min_{i \in [n]} q_i^2}, \left(\frac{12 k_6 \sqrt{d+1} \log T}{\gamma \Delta_0 \min_{i \in [n]} q_i^{\kappa/2}} \right)^{4/(2\kappa-1)}, \left(\frac{12 k_6}{\gamma \min_{i \in [n]} q_i^{\kappa/2}} \right)^{2/\kappa} \right\}, \tag{2.36}$$

where k_6 is some constant, we have that $\hat{\mathcal{N}}_t = \mathcal{N}_{i_t}$ with probability at least $1 - 2n/t^2$.

Proof: We consider the estimation error of β_i and α_i, and we want to show that both of them can be controlled. According to Lemma 2.6.7, if $t > 4\log(2t)/\min_{i \in [n]} q_i^2$, we have that for any $i \in [n]$ $T_{i,t} \geq q_i t/2$ with probability at least $1 - 1/t^2$ (since $D(t) < q_i t/2$ for all $i \in [n]$). If this this true, we have $\sum_{s \in \mathcal{T}_{i,t}} \Delta_s^2 \geq \Delta_0^2 T_{i,t}/\sqrt{t} \geq \Delta_0^2 q_i \sqrt{t}/2$. Moreover, because of Assumption B.2 and $t > 2t_0/\min_i q_i$ (which implies that $T_{i,t} > t_0$ for all $i \in [n]$), $\lambda_{\min}(V_{i,t}) \geq c_0 T_{i,t}^{\kappa}$. As a result,

$$\begin{aligned} C_{i,t} \leq & k_5 \left(\sqrt{d+1} \log t \sqrt{\frac{2T_{i,t}}{\Delta_0^2 q_i \sqrt{t} \lambda_{\min}(V_{i,t})}} + \sqrt{\frac{\log t}{\Delta_0^2 q_i \sqrt{t}}} + \sqrt{\frac{1}{\lambda_{\min}(V_{i,t})}} \right) \\ \leq & k_6 \left(\sqrt{d+1} \log t \sqrt{\frac{t^{1/2-\kappa}}{\Delta_0^2 q_i^{\kappa}}} + \sqrt{\frac{\log t}{\Delta_0^2 q_i \sqrt{t}}} + \sqrt{(q_i t)^{-\kappa}} \right) \end{aligned}$$

for some constant k_5, k_6 with probability at least $1 - 1/t^2$. Since Lemma 2.6.9 implies that $||\hat{\theta}_{i,t} - \theta_i||_2 \leq C_{i,t}$ with probability at least $1 - 1/t^2$, if

$$t > \max \left\{ \left(\frac{12 k_6 \sqrt{d+1} \log t}{\gamma \Delta_0 \min_{i \in [n]} q_i^{\kappa/2}} \right)^{4/(2\kappa-1)}, \left(\frac{12 k_6}{\gamma \min_{i \in [n]} q_i^{\kappa/2}} \right)^{2/\kappa} \right\},$$

we have $||\hat{\theta}_{i,t} - \theta_i||_2 \leq C_{i,t} < \gamma/4$ for all $i \in [n]$ with probability at least $1 - 2n/t^2$. Then using Lemma 2.6.5 leads to the result. □

Lemma 2.6.11 *For any $t > 2\bar{t}'$, we have that*

$$|\tilde{\beta}_{j_t,t} - \beta_{j_t}| \leq k_8\sqrt{\log t}(\tilde{q}_{j_t}t)^{-1/4}/\Delta_0$$
$$|\tilde{\alpha}'_{j_t,t}x - \alpha'_{j_t}x| \leq k_7\sqrt{(d+1)}\log t(\tilde{q}_{j_t}t)^{1/4}/\Delta_0||x||_{\tilde{V}^{-1}_{j_t,t}}$$

for some constants k_7, k_8 with probability at least $1 - 7n/t$.

Proof: According to Corollary 2.6.2 and Cauchy-Schwarz, we have

$$\begin{aligned}|\tilde{\beta}_{j_t,t} - \beta_j| \leq & k_4\sqrt{2\log(1+t)}\left(\sum_{s\in\tilde{\mathcal{T}}_{j_t,t}}\Delta_s^2\right)^{-1/2} \\ |\tilde{\alpha}'_{j_t,t}x - \alpha'_j x| \leq & k_3\sqrt{d+1}2\log(1+t)\left(\sum_{s\in\tilde{\mathcal{T}}_{j_t,t}}\Delta_s^2\right)^{-1/2}\sqrt{\tilde{T}_{j_t,t}}||x||_{\tilde{V}^{-1}_{j_t,t}}\end{aligned} \tag{2.37}$$

with probability at least $1 - 1/t$.

Define events $\mathcal{E}_{N,s} = \{\hat{\mathcal{N}}_s = \mathcal{N}_{i_s}\}$. According to Lemma 2.6.10, when $s > \bar{t}'$, $\mathcal{E}_{N,s}$ holds with probability at least $1 - 2n/s^2$. Note that on events $\mathcal{E}_{N,s}$ for all $s \in [t/2, t]$ (which holds with probability at least $1 - 4n/t$ as $t/2 > \bar{t}'$), we have that

$$\sum_{s\in\tilde{\mathcal{T}}_{j_t,t}}\Delta_s^2 \geq \sum_{s\in\tilde{\mathcal{T}}_{j_t,t}:s>t/2}\Delta_s^2 \geq \sum_{s\in\tilde{\mathcal{T}}_{j_t,t}:s>t/2}\frac{\Delta_0^2|\tilde{\mathcal{T}}_{j_t,t}\cap\{s>t/2\}|}{\sqrt{\tilde{T}_{j_t,t}}}.$$

Then according to Lemma 2.6.8, $|\tilde{\mathcal{T}}_{j_t,t}\cap\{s > t/2\}| \in [\tilde{q}_{j_t}t/2 - \tilde{D}(t/2), \tilde{q}_{j_t}t/2 + \tilde{D}(t/2)]$ where $\tilde{D}(t/2) = \sqrt{t\log(2t)/2} \leq \tilde{q}_{j_t}t/4$ (because $t > 2\bar{t}'$) with probability at least $1 - 1/t$ (hence $|\tilde{\mathcal{T}}_{j_t,t}\cap\{s > t/2\}| \geq \tilde{q}_{j_t}t/4$). Similarly, we also have $\tilde{T}_{j_t,t} \in [\tilde{q}_{j_t}t/2, 3\tilde{q}_{j_t}t/2]$ with probability at least $1 - 1/t$. As a result, combined with the above equation, with probability at least $1 - 6n/t$, we have

$$\sum_{s\in\tilde{\mathcal{T}}_{j_t,t}}\Delta_s^2 \geq \frac{\Delta_0^2\sqrt{\tilde{q}_{j_t}t}}{4}.$$

Combining with (2.37), we obtain the desired result. □

2.6.3 Different demand parameters for the same cluster

As mentioned in Remark 1 in Section 2.3, this section talks about some technical lemmas in showing the regret of the modified CSMP when parameters θ_i within the same cluster can be different. Note that we assume $||\theta_{i_1} - \theta_{i_2}||_2 \leq \gamma_0$ for any i_1, i_2 in any cluster $\mathcal{N}_j$.

The first result is an corollary of Lemma 2.6.5.

Corollary 2.6.3 *Suppose for all $i \in [n]$ it holds that $||\hat{\theta}_{i,t-1} - \theta_i||_2 \leq B_{i,t-1}$ and $B_{i,t-1} < \gamma/8$. Then in the modified algorithm (with $\gamma > 2\gamma_0$), we have that $\hat{\mathcal{N}}_t = \mathcal{N}_{i_t}$.*

Proof: The proof is almost identical to Lemma 2.6.5. First of all, for $i_1, i_2 \in [n]$, if they belong to different clusters and $B_{i_1,t-1}+B_{i_2,t-1} < \gamma/4$, we must have $||\hat{\theta}_{i_1,t-1}-\hat{\theta}_{i_2,t-1}||_2 > B_{i_1,t-1} + B_{i_2,t-1} + \gamma_0$ because

$$\begin{aligned}\gamma \leq & ||\theta_{i_1} - \theta_{i_2}||_2 \leq ||\theta_{i_1} - \hat{\theta}_{i_1,t-1}||_2 + ||\hat{\theta}_{i_1,t-1} - \hat{\theta}_{i_2,t-1}||_2 + ||\hat{\theta}_{i_2,t-1} - \theta_{i_2}||_2 \\ \leq & B_{i_1,t-1} + ||\hat{\theta}_{i_1,t-1} - \hat{\theta}_{i_2,t-1}||_2 + B_{i_2,t-1} < \gamma/4 + ||\hat{\theta}_{i_1,t-1} - \hat{\theta}_{i_2,t-1}||_2,\end{aligned}$$

which implies that $||\hat{\theta}_{i_1,t-1} - \hat{\theta}_{i_2,t-1}||_2 > 3\gamma/4 > \gamma/4 + \gamma_0 > B_{i_1,t-1} + B_{i_2,t-1} + \gamma_0$.

On the other hand, if $||\hat{\theta}_{i_1,t-1} - \hat{\theta}_{i_2,t-1}||_2 > B_{i_1,t-1} + B_{i_2,t-1} + \gamma_0$, we must have i_1, i_2 belongs to different clusters because

$$\begin{aligned}B_{i_1,t-1} + B_{i_2,t-1} + \gamma_0 < & ||\hat{\theta}_{i_1,t-1} - \hat{\theta}_{i_2,t-1}||_2 \\ \leq & ||\theta_{i_1} - \hat{\theta}_{i_1,t-1}||_2 + ||\theta_{i_1,t-1} - \theta_{i_2,t-1}||_2 + ||\hat{\theta}_{i_2,t-1} - \theta_{i_2}||_2 \\ \leq & B_{i_1,t-1} + ||\theta_{i_1,t-1} - \theta_{i_2,t-1}||_2 + B_{i_2,t-1}\end{aligned}$$

which implies $||\theta_{i_1,t-1} - \theta_{i_2,t-1}||_2 > \gamma_0$, i.e., they belong to different clusters.

Therefore, if $i \in \hat{\mathcal{N}}_t$, i.e., $||\hat{\theta}_{i_t,t-1} - \hat{\theta}_{i,t-1}|| \leq B_{i_t,t-1} + B_{i,t-1} + \gamma_0$, we must have that $i \in \mathcal{N}_{i_t}$ as well or $B_{i_t,t-1} + B_{i,t-1} \geq \gamma/4$ (which is impossible by our assumption that $B_{i,t-1} < \gamma/8$).

On the other hand, if $i \in \mathcal{N}_{i_t}$, then we must have $||\hat{\theta}_{i_t,t-1}-\hat{\theta}_{i,t-1}|| \leq B_{i_t,t-1}+B_{i,t-1}+\gamma_0$, which implies that $i \in \hat{\mathcal{N}}_t$ as well. Summarizing, we have shown that $\hat{\mathcal{N}}_{i_t} = \mathcal{N}_{i_t}$. □

The next lemma measures the confidence bound of $\tilde{\theta}_{j,t}$ compared with any true parameter $\tilde{\theta}_i$ for $i \in \mathcal{N}_j$, with respect to the empirical Fisher's information matrix $\tilde{V}_{j,t}$.

Lemma 2.6.12 *Let t satisfies that*

$$t > \left(\frac{8R\log((d+2)T)}{\lambda_1 \Delta_0^2 \min_j \tilde{q}_j}\right)^2.$$

On the event that $\tilde{T}_{j_t,t} \geq \tilde{q}_{j_t} t/2$*,*

$$||\tilde{\theta}_{j_t,t} - \bar{\theta}_{j_t}||_2 \leq \frac{2\sqrt{(d+2)\log\left(1 + tR^2/(d+2)\right) + 4\log t} + 2l_1 L}{l_1 \sqrt{\lambda_{\min}(\tilde{V}_t)}} + \frac{2L_1 R^2 \gamma_0}{l_1 \lambda_1 \Delta_0^2 \upsilon^2}$$

with probability at least $1 - 2/t^2$*.*

Proof: The proof is quite similar to Lemma 2.6.2. We drop the index j_t for convenience. Note that for an arbitrary parameter $\phi \in \Theta$, since $\tilde{\theta}_t$ is the MLE, we have

$$\begin{aligned} 0 \geq \sum_s l_s(\tilde{\theta}_t) - \sum_s l_s(\phi) = \sum_s \nabla l_s(\phi)'(\tilde{\theta}_t - \phi) + \frac{1}{2}\sum_s \dot{\mu}(u_s'\bar{\phi}_t)(u_s'(\tilde{\theta}_t - \phi))^2 \\ + \frac{l_1}{2}||\tilde{\theta}_t - \phi||_2^2 - \frac{l_1}{2}||\tilde{\theta}_t - \phi||_2^2 \geq \sum_s \nabla l_s(\phi)'(\tilde{\theta}_t - \phi) + \frac{l_1}{2}||\tilde{\theta}_t - \phi||_{\tilde{V}_t}^2 - 2l_1 L^2, \end{aligned} \tag{2.38}$$

where the first inequality is from the optimality of $\tilde{\theta}_t$, and $\overline{\phi}_t$ is a point on line segment between $\tilde{\theta}_t$ and ϕ.

Now we consider $\nabla l_s(\phi)$. By Taylor's theorem, $\nabla l_s(\phi) = \nabla l_s(\theta_s) + \nabla^2 l_s(\check{\theta}_s)'(\phi - \theta_s)$, where θ_s is the true parameter at time s, and $\check{\theta}_s$ is a point between ϕ and θ_s. As a result,

$$\nabla l_s(\phi) = -\epsilon_s u_s + \dot{\mu}(u_s'\check{\theta}_s) u_s u_s'(\phi - \theta_s). \tag{2.39}$$

Since $\phi \in \Theta$ is an arbitrary vector, we can let $\phi = \theta_i$ for any $i \in \mathcal{N}_j$. Combining (2.38)

and (2.39), we have that with probability at least $1 - 1/t^2$.

$$\begin{aligned}
\frac{l_1}{2}||\tilde{\theta}_t - \theta_i||^2_{\tilde{V}_t} \leq & \sum_s \epsilon_s u_s'(\tilde{\theta}_t - \theta_i) - \sum_s \dot{\mu}(u_s'\check{\theta}_s)(\theta_i - \theta_s)' u_s u_s'(\tilde{\theta}_t - \phi) + 2l_1 L^2 \\
\leq & ||\sum_s \epsilon_s u_s||_{\tilde{V}_t^{-1}} ||\tilde{\theta}_t - \theta_i||_{\tilde{V}_t} + \sum_s ||\dot{\mu}(u_s'\check{\theta}_s) u_s u_s'(\theta_i - \theta_s)||_{\tilde{V}_t^{-1}} ||\tilde{\theta}_t - \theta_i||_{\tilde{V}_t} \\
& + 2l_1 L^2 \\
\leq & \sqrt{(d+2)\log\left(1 + \frac{tR^2}{d+2}\right) + 4\log t}||\tilde{\theta}_t - \theta_i||_{\tilde{V}_t} \\
& + \frac{\sum_s ||\dot{\mu}(u_s'\check{\theta}_s) u_s u_s'(\theta_i - \theta_s)||_2 ||\tilde{\theta}_t - \theta_i||_{\tilde{V}_t}}{\sqrt{\lambda_{\min}(\tilde{V}_t)}} + 2l_1 L^2 \\
\leq & \sqrt{(d+2)\log\left(1 + \frac{tR^2}{d+2}\right) + 4\log t}||\tilde{\theta}_t - \theta_i||_{\tilde{V}_t} + \frac{L_1 R^2 \gamma_0 \tilde{q}_j t||\tilde{\theta}_t - \theta_i||_{\tilde{V}_t}}{2\sqrt{\lambda_{\min}(\tilde{V}_t)}} \\
& + 2l_1 L^2,
\end{aligned}$$

where the second inequality is from Theorem 1 in Abbasi-Yadkori et al. (2011) and the last inequality is because $\tilde{T}_{j_t,t} \geq \tilde{q}_{j_t} t/2$. By some simple algebra, above inequality implies that

$$||\tilde{\theta}_t - \theta_i||_{\tilde{V}_t} \leq \frac{2\sqrt{(d+2)\log\left(1 + \frac{tR^2}{d+2}\right) + 4\log t}}{l_1} + \frac{L_1 R^2 \gamma_0 \tilde{q}_j t}{l_1 \sqrt{\lambda_{\min}(\tilde{V}_t)}} + 2L.$$

This inequality further implies that

$$||\tilde{\theta}_t - \theta_i||_2 \leq \frac{2\sqrt{(d+2)\log\left(1 + \frac{tR^2}{d+2}\right) + 4\log t}}{l_1 \sqrt{\lambda_{\min}(\tilde{V}_t)}} + \frac{L_1 R^2 \gamma_0 \tilde{q}_j t}{l_1 \lambda_{\min}(\tilde{V}_t)} + \frac{2L}{\sqrt{\lambda_{\min}(\tilde{V}_t)}}. \quad (2.40)$$

Since in the modified algorithm, we let $\Delta_t = \pm \Delta_0 \max\left(\tilde{T}_{\tilde{\mathcal{N}}_t,t}^{-1/4}, \upsilon\right)$, on the event d Lemma 2.6.4 implies that $\lambda_{\min}(\tilde{V}_t) \geq \lambda_1 \Delta_0^2 \tilde{q}_j \max(\sqrt{t}, \upsilon^2 t)/2 \geq \lambda_1 \Delta_0^2 \upsilon^2 \tilde{q}_j t/2$ with probability at least $1 - 1/t^2$ for any t satisfying $t > \left(8R\log((d+2)T)/(\lambda_1 \Delta_0^2 \min_j \tilde{q}_j)\right)^2$. Plug $\lambda_{\min}(\tilde{V}_t) \geq \lambda_1 \Delta_0^2 \upsilon^2 \tilde{q}_j t/2$ in $L_1 R^2 \gamma_0 \tilde{q}_j t/(l_1 \lambda_{\min}(\tilde{V}_t))$ in (2.40), we finally show that with probability at least $1 - 2/t^2$,

$$||\tilde{\theta}_t - \theta_i||_2 \leq \frac{2\sqrt{(d+2)\log\left(1 + \frac{tR^2}{d+2}\right) + 4\log t} + 2l_1 L}{l_1 \sqrt{\lambda_{\min}(\tilde{V}_t)}} + \frac{2L_1 R^2 \gamma_0}{l_1 \lambda_1 \Delta_0^2 \upsilon^2},$$

and we finish the proof. □ Now we provide the proof (sketch) of the theorem of regret of modified algorithm.

Theorem 2.6.1 *The expected regret of the modified algorithm CSMP is*

$$R(T) = O\left(\frac{d^2\log^2(dT)}{\min_{i\in[n]} q_i^2} + d\log T\sqrt{mT} + \gamma_0^{2/3}T\right).$$

If we hide logarithmic terms and let $\min_{i\in[n]} q_i = \Theta(1/n)$ *with* $T \gg n$*, we have the expected regret is at most* $R(T) = \tilde{O}(d\sqrt{mT} + \gamma_0^{2/3}T)$.

Proof: The proof is almost identical to Theorem 2.3.1 so we neglect most part of the proof. The only thing which requires extra investigation is that conditioned on various events as in Theorem 2.3.1, and let t sufficiently large (larger than some time with the same scale as the maximum of $\bar{t}$), we want to bound $r_t(p_t^*) - r_t(p_t) = O(r_t(p_t^*) - r_t(p_t') + \Delta_t^2)$. Note that $\Delta_t^2 = O\left(\max\left(\tilde{T}_{\hat{\mathcal{N}}_t,t}^{-1/2}, \upsilon^2\right)\right) \leq O\left(\tilde{T}_{\hat{\mathcal{N}}_t,t}^{-1/2} + \upsilon^2\right)$, and for the part of regret $\sum_t O\left(\tilde{T}_{\hat{\mathcal{N}}_t,t}^{-1/2}\right)$, it is bounded as in Theorem 2.3.1. From υ^2, the cumulative regret becomes $O(\upsilon^2 T)$.

To bound $r_t(p_t^*) - r_t(p_t')$, note that we have $r_t(p_t^*) - r_t(p_t') \leq O\left(||\theta_{i_t} - \tilde{\theta}_{j_t,t}||_2^2\right)$. Now we use the result in Lemma 2.6.12 and obtain that

$$r_t(p_t^*) - r_t(p_t') \leq O\left(\frac{d\log t}{\lambda_{\min}(\tilde{V}_t)} + \frac{\gamma_0^2}{\upsilon^4}\right).$$

The cumulative regret by summing over $O(d\log t/\lambda_{\min}(\tilde{V}_t))$ is the same as in Theorem 2.3.1, and the cumulative regret from $O(\gamma_0^2/\upsilon^4)$ is obviously $O(\gamma_0^2 T/\upsilon^4)$.

Above all, adding up all parts of regret, we have that the expected regret is at most

$$R(T) = O\left(\frac{d^2\log^2(dT)}{\min_{i\in[n]} q_i^2} + d\log T\sqrt{mT} + \upsilon^2 T + \frac{\gamma_0^2 T}{\upsilon^4}\right).$$

Taking value $\upsilon = \Theta(\gamma_0^{1/3})$ gives us the final result. □

2.7 Conclusion

With the rapid development of e-commerce, data-driven dynamic pricing is becoming increasingly important due to the dynamic market environment and easy access to online sales data. While there is abundant literature on dynamic pricing of normal products, the pricing of products with low sales received little attention. The data from Alibaba Group

shows that the number of such low-sale products is large, and that even though the demand for each low-sale product is small, the total revenue for all the low-sale products is quite significant. In this chapter, we present data clustering and dynamic pricing algorithms to address this challenging problem. We believe that this chapter is the first to integrate online clustering learning in dynamic pricing of low-sale products.

Two learning algorithms are developed in this chapter: one for a dynamic pricing problem with the generalized linear demand, and another for the special case of linear demand functions under weaker assumptions on product covariates. We have established the regret bounds for both algorithms under mild technical conditions. Moreover, we test our algorithms on a real dataset from Alibaba Group by simulating the demand function. Numerical results show that both algorithms outperform the benchmarks, where one either considers all products separately, or treats all products as a single cluster. A field experiment was conducted at Alibaba by implementing the CSMP algorithm on a set of products, and the results show that our algorithm can significantly boost revenue.

There are several possible future research directions. The first one is an in-depth study of the method for product clustering. For instance, in bandit clustering literature, Gentile et al. (2014) use a graph-based method to cluster different arms, and Nguyen and Lauw (2014) apply a K-means clustering method to identify different groups of arms. It will be interesting to understand the various product clustering methods and analyze their advantages and disadvantages under different scenarios. Second, to highlight the benefit of clustering techniques for low-sale products, in this chapter we study a dynamic pricing problem with sufficient inventory. One extension is to apply the clustering method for the revenue management problem with inventory constraint. Third, in this chapter we consider the generalized linear demand. There are other general demand functions, such as the nonparametric models in Araman and Caldentey (2009), Wang et al. (2014), Chen et al. (2015a), Besbes and Zeevi (2015), Nambiar et al. (2018), Ferreira et al. (2018a), Chen and Gallego (2018), and it is an interesting research direction to explore other, and broader, classes of demand functions. To that end, an important step will be to define an appropriate metric for clustering the products, which is a challenge especially for nonparametric models. In the end, we believe that it will be interesting to include substitutability/complementarity of products and even assortment decisions.

Chapter 3

Online Personalized Assortment Optimization in a Big Data Regime

3.1 Introduction

With the advancement of information technology, customization has become increasingly important for e-business. Many internet firms face the problem of recommending items, called an *assortment*, to a customer tailored to his/her personal preference. For example, in online retailing, whenever a customer clicks into a website, e.g., Amazon.com, the customer is shown a set of products that are generated according to that customer's historical clicking/purchasing records; in online video streaming websites like Youtube and Netflix, a selection of videos are recommended to each customer on his/her homepage based on that customer's viewing history; in social media websites, e.g., Facebook, advertisement is posted according to the user's browsing records; in news websites such as Yahoo!, personalized recommendations of articles are made to each reader. The problem of recommending an assortment of items to a customer according to his/her preference using personal data is known as *personalized assortment optimization*. A number of success stories have been reported on the implementation of personalized assortment optimization algorithms. For instance, according to Netflix executives Carlos A. Gomez-Uribe and Neil Hunt, their recommendation system saves Netflix over $\$1$ billion each year (Gomez-Uribe and Hunt 2016), and a Microsoft Research report estimated that 30% of Amazon.com's page views during a 10-month period were from personalized recommendation (Sharma et al. 2015).

A great deal of research have been done on recommendation systems (see e.g., Bobadilla et al. 2013 for a comprehensive survey); each focuses on a specific class of problem with restricted applications. The machine learning methods proposed for recommendation systems, e.g., collaborative filtering (Herlocker et al. 2000) and deep learning (Covington et al. 2016), typically require abundant training data, which may not be available in dynamic online settings when new items (e.g., new fashion designs, new uploaded videos)

are introduced frequently to the market. Personalized assortment optimization requires to recommend a set of items, but most existing methods are concerned with single item recommendation that give a score to each item and choose the "top K" items with highest scores (see e.g., Sarwar et al. 2001, Davidson et al. 2010, Cremonesi et al. 2010). It is very likely that there exist substitution effects among the items recommended, and failing to capture these effects can lead to suboptimal solutions (see e.g., Feldman et al. 2018 for a recent field experiment result at Alibaba for comparing choice-based model with single-product recommendation algorithms). There are several recent papers on *online personalized assortment optimization*, e.g., Cheung and Simchi-Levi (2017) and Chen et al. (2018a), that study a similar problem as ours but their algorithms suffer from computational inefficiency. More specifically, as data accumulates, the computational time in each round of their algorithms increases. This makes it difficult to make fast decisions to each arriving customer in real time and limits their applications. For instance, according to Yahoo! Front Page Today Module User Click Log Data, the number of customer clicks during one time interval of the day on May 4th, 2009 was already more than 5,000,000. Moreover, the data of each user might have high dimension, slowing down the computation even further.

In this chapter, we address the issues discussed above by developing efficient algorithms for online personalized assortment optimization problem when customer choice parameters are not known *a priori*. Demand learning mechanism is designed that can handle big (and high dimensional) data, thus it is amenable for applications in a big data regime. The theoretical performances of the algorithms are shown to be near-optimal, and numerical results based on real data outperform benchmark algorithms.

The decision process of our online personalized assortment optimization problem proceeds as follows: N products are sold over a time horizon of T periods. Each period $t = 1, 2, \ldots$ represents an arrival that has an observable information data (e.g., profile data, clicking/purchasing historical data), denoted by $x_t \in \mathbb{R}^D$, where D is the dimension of the vector which can be large. Based on the personalized information x_t, the firm selects an assortment of at most K items to display to the customer. Observing the set of products on display, the customer makes a purchasing decision following a multinomial logit (MNL) choice model. Not knowing the parameters of the choice model *a priori*, the firm wants to maximize the expected total reward (e.g., clicking-through rate, revenue) over the time horizon.

3.1.1 Main contribution of the chapter

We design algorithms which simultaneously learn the demand and determine the assortment on the fly, such that the total reward is maximized. The algorithms are easy to implement with efficient computation no matter how much (possibly high dimensional) data have been accumulated in a period. The algorithms are shown to perform very well in terms of regret, defined as the total revenue loss compared with a clairvoyant who has complete demand information *a priori* and always makes the best decision.

The main contributions in this chapter are summarized as follows.

- Our first learning algorithm, which we refer to as P-UCB, is designed based on maximum likelihood estimation and a personalized upper-confidence bound (UCB). The personalized UCB allows us to simultaneously learn the parameters of different products and maximize the earned revenue. We show in Theorem 3.3.1 that the regret of this algorithm is at most $\tilde{O}(DNK\sqrt{T})$, where the notation $\tilde{O}(\cdot)$ hides the logarithmic terms.

- To resolve the issue of slowing computation as demand data accumulate in a typical demand learning problem, we develop another algorithm, called OLP-UCB which is based on P-UCB, by incorporating an online convex optimization scheme to update the estimated parameters. This algorithm has constant computational effort in each iteration that is independent of the time period; thus its computation time does not increase in the accumulated data size and is particularly useful when solving online personalized assortment optimization problem in a big data regime. We prove in Theorem 3.3.3 that the regret of OLP-UCB is at most $\tilde{O}(DNK^{3/2}\sqrt{T})$, which is only slightly higher than that of the first algorithm.

- We present the third algorithm, called OLP-UCB-RP, for high dimensional personalized data setting. Assuming some sparsity structure of the high dimensional customers' data, which is prevalent in applications, we introduce a random projection step for dimension reduction. As the dimension reduction step can usually be conducted offline (the decision maker typically has access to the database), the OLP-UCB-RP algorithm drastically speeds up the computation. With a significant reduction in computational cost, we prove that OLP-UCB-RP algorithm still achieves satisfactory theoretical performance: Theorem 3.4.1 shows that its regret is at most $\tilde{O}(NK^{3/2}\sqrt{(d_0+d)LT}+(d_0+d)NK^{3/2}\sqrt{T})$, where d_0 and L are parameters related to the data sparsity to be specified later, and d (which is much smaller than D) is the dimension of data after random projection.

- Our algorithms are tested on a real dataset from Yahoo! on news article recommendation, and unbiased estimates are obtained on the clicking through rate (CTR). The results show a 39.73% increase in CTR over Yahoo!'s current recommendation method. We also test the algorithms using a synthetic data and the numerical results demonstrate excellent performance.

We note that similar problem has been studied in Cheung and Simchi-Levi (2017) and Chen et al. (2018a) (the latter is done concurrently and independently of ours). This chapter has significant differences from those studies. First, Cheung and Simchi-Levi (2017) present a Thompson Sampling algorithm and evaluate the algorithm using Bayesian regret, a weaker measure than regret (see e.g., Russo and Van Roy 2014), and their Bayesian regret is $\tilde{O}(DNK^{5/2}\sqrt{T})$; in this chapter we develop a learning algorithm and evaluate it using regret, and our result is $\tilde{O}(DNK\sqrt{T})$ for P-UCB. Second, unlike these references that have linearly increasing computational time in period index, our OLP-UCB algorithm has a constant computational time in each period. Third, compared with the references, our OLP-UCB-RP algorithm can be used to solve problems with high dimensional data. Fourth, our algorithms and results do not require any stochastic assumption on data, i.e., the sequence of data can be arbitrary or even adversarial, while Chen et al. (2018a) requires stochastic assumption on (some of) personal information data. Finally, the assortment in each iteration of our algorithm is computed using exact and efficient method, while the selection of assortment in Chen et al. (2018a) has to rely on approximation algorithm, since the optimization problem they formulate is too complex to adopt an exact method to find an optimal solution.

3.1.2 Related literature

In this section, we briefly review some related research from both the operations management and the machine learning literature.

Related literature on assortment optimization. Assortment optimization has been an important research area for decades. Earlier work on this topic has mainly focused on the static optimization problem (see e.g., Ryzin and Mahajan 1999, Mahajan and Van Ryzin 2001, Gaur and Honhon 2006, Cachon and Kök 2007, Davis et al. 2014, Gallego and Topaloglu 2014, Rusmevichientong et al. 2014, Li et al. 2015.). We refer interested readers to Kök et al. (2015) for a comprehensive literature review. In recent years, due to the popularity of data-driven revenue management problems, dynamic assortment with demand learning has become increasingly popular. To the best of our knowledge, Caro and Gallien (2007) is the first to study this type of problem, though they assume that demands for differ-

ent products are independent of each other. Another popular demand model for assortment optimization is the so-called *multinomial logit* (MNL) choice model, where customers' choices are modeled by the perceived product utilities, which leads to demands of products in the assortment to be dependent. Several papers have used MNL model to study the dynamic assortment optimization with demand learning, see e.g., Rusmevichientong et al. (2010), Sauré and Zeevi (2013), Agrawal et al. (2017a,b), Wang et al. (2018); a slightly more general model called *nested logit* (NL) has been studied, as well (Chen et al. 2018b).

The papers cited above assume homogeneous customers. Works on dynamic assortment optimization with heterogeneous customers (i.e., online personalized assortment optimization) include Golrezaei et al. (2014), Chen et al. (2015b), Bernstein et al. (2015), Kallus and Udell (2016), Gallego et al. (2016), Cheung and Simchi-Levi (2017), Bernstein et al. (2018), Chen et al. (2018a). Among these papers, Golrezaei et al. (2014), Bernstein et al. (2015), Kallus and Udell (2016), Gallego et al. (2016), Bernstein et al. (2018) model the heterogeneity of customers by customer segmentation. In particular, Golrezaei et al. (2014), Bernstein et al. (2015), Gallego et al. (2016) study personalized assortment optimization with initial capacity constraint and known demand information, and develop heuristics with provable theoretical performance. In Kallus and Udell (2016) and Bernstein et al. (2018), the authors study the problem of demand learning, and propose learning algorithms tailored for different customer segments. Chen et al. (2015b), Cheung and Simchi-Levi (2017), Chen et al. (2018a) represent personal information data of each arriving customer using vector, which is the formulation we adopt in this chapter. The differences between these papers and ours have been discussed in the the previous subsection.

Related literature on multi-armed bandit problem. *Multi-armed bandit* (MAB) problems have received much attention in the literature, and a very important method is called upper-confidence bound method (UCB, see e.g., Auer 2002). One area that is closely related to ours is *contextual bandits* (see e.g., Zhou 2015, for a comprehensive survey), which is an important topic in multi-armed bandit problem. Most of the research in this area focuses on the linear contextual bandit problem (in which the objective function is a linear function of context), such as Auer (2002), Dani et al. (2008), Rusmevichientong and Tsitsiklis (2010), Chu et al. (2011), Abbasi-Yadkori et al. (2011), Agrawal and Goyal (2013). Some recent work extends results to the *generalized linear bandit* (Filippi et al. 2010, Li et al. 2017b, Jun et al. 2017). For the generalized linear bandit problem, the objective function is a generalized linear function of the context, and the logistic function is a special case. Although the MNL model is a generalization of the logistic function, our problem is different from the contextual bandit problem because we choose an assortment of items instead of a single arm. This combinatorial choice of items leads to significantly

more complicated objective function in our problem than in generalized linear bandit. As a result, the methods in contextual bandit literature cannot be directly applied in our problem.

One scenario considered in this chapter is the high dimensional online personalized assortment optimization. There are some research on contextual bandits with high dimensional data (Abbasi-Yadkori et al. 2012, Carpentier and Munos 2012, Bastani and Bayati 2015). In these papers, the underlying unknown parameter is assumed to be sparse. This assumption is difficult to verify because the decision maker does not know the underlying parameters. This chapter therefore takes a different approach, assuming that the customer data is sparse. This assumption is observable from the dataset and prevalent in reality (see e.g., our data analysis of a real dataset from Alibaba in Section 3.4).

3.1.3 Organization of the chapter

The remainder of this chapter is organized as follows. In Section 3.2 we introduce the problem formulation and some basic assumptions. We develop our algorithms (P-UCB and OLP-UCB) and present their theoretical performance in Section 3.3. We extend the algorithms to high dimensional data setting using random projection in Section 3.4. Numerical experiments of our algorithms are presented in Section 3.5, and several benchmark algorithms are compared with our algorithms. In Section 3.6, we outline the main steps of the proofs of the theorems, with technical details provided in the Section 3.7. Finally, we conclude the chapter with some discussion and future research directions in Section 7.

3.2 Problem Formulation

A firm sells N products, labeled as $i = 1, 2, \ldots, N$, over T periods. Denote the set of time periods by $\mathcal{T}$. The selling price (or reward) of product i is $p_i > 0$, which is exogenous. During each period t, the firm can display up to K products, called an assortment. As described in the previous section, there is exactly one arrival (customer) in a period, that either purchases one of the products on display or leaves without purchasing any product. Each customer is associated with an observable personal (contextual) information vector represented by $x_t \in \mathbb{R}^D$. We do not make any stochasticity assumption on x_t; i.e., they can be arbitrary and might even depend on the assortment decisions in earlier periods. The firm needs to determine, in each period, the assortment of products to offer, using customer's information x_t, to maximize the expected total revenue over the time horizon $\mathcal{T}$.

For convenience, we denote the set of all products by $\mathcal{N} := \{1, 2, \ldots, N\}$, and the collection of all possible assortments by $\mathcal{S} := \{S \subset \mathcal{N} : \ |S| \leq K\}$. Here, and in what

follows, ":=" stands for "defined as", and $|S|$ denotes the number of elements in, or the cardinality of, assortment S.

We adopt the widely accepted multinomial logit (MNL) model for customer's choice (see Feldman et al. 2018 for a field experiment result of assortment optimization using MNL). Under MNL model, the probability for a customer with personal data x_t to choose product $i \in S \cup \{0\}$, with 0 denoting the option of purchasing nothing, is

$$q(i|S, x_t) := \frac{e^{x_t'\theta_i}}{1 + \sum_{j \in S} e^{x_t'\theta_j}}, \qquad \forall i \in S,$$
$$q(0|S, x_t) := \frac{1}{1 + \sum_{j \in S} e^{x_t'\theta_j}},$$

where the parameters θ_i for $i \in \mathcal{N}$ are D-dimensional column vectors that are unknown to the firm *a priori*, and need to be learned through sales data. Given the assortment S, the expected revenue from a customer with information x_t is

$$r(S, x_t) := \sum_{i \in S} p_i q(i|S, x_t). \tag{3.1}$$

To emphasize its dependency on demand parameters $\theta' := (\theta_1', \ldots, \theta_N')$, where θ_i' is the transpose of (column) vector θ_i, we sometimes also write the revenue function as $r(S, x_t, \theta)$. By normalization, we assume without loss of generality that $||x_t||_2 \leq 1$ for all $t \in \mathcal{T}$ and $||\theta_i||_2 \leq R$ for all $i \in \mathcal{N}$, where R is a positive constant. For convenience, we define the feasible set of θ as $\Theta = \bigotimes_{i=1}^N \Theta_i$ where $\Theta_i := \{\theta_i \in \mathbb{R}^D : ||\theta_i||_2 \leq R\}$. For the convenience of subsequent discussion, we introduce notations

$$\overline{\kappa} := e^R, \underline{\kappa} := e^{-R}. \tag{3.2}$$

Then, $e^{x_t'\theta_i} \in [\underline{\kappa}, \overline{\kappa}]$ for any x_t and $\theta_i \in \Theta_i$. Finally, we will call a real number *a constant* if it depends only on R and $\max_{i \in \mathcal{N}} p_i$, but not on the specific values of θ_i and p_i.

Clairvoyant solution and the firm's objective. If the demand parameter θ is known *a priori*, then the firm can choose an optimal $S_t^* \in \mathcal{S}$ which maximizes the revenue function (3.1). We call this optimal solution a *clairvoyant solution*, which can be computed efficiently using algorithms in the existing literature (see e.g., Rusmevichientong et al. 2010, Davis et al. 2013, but see Remark 1 below). Again to emphasize its dependency on θ and x_t, we will at times write it as $S_t^* = S^*(\theta, x_t)$ when necessary. The clairvoyant revenue

over the time horizon is

$$J^*(T) := \sum_{t=1}^{T} r(S_t^*, x_t),$$

and it will be used as a benchmark to analyze the performance of our learning algorithm.

Remark 3.2.1 *When customer information x_t depends on earlier assortment decision, the clairvoyant solution $S^*(\theta, x_t)$ is the optimal solution to the Markov decision process. In this chapter we follow the literature to take the clairvoyant solution $S^*(\theta, x_t)$ as the myopic optimal solution (see e.g., Auer 2002, Dani et al. 2008, Rusmevichientong and Tsitsiklis 2010, Chu et al. 2011, Abbasi-Yadkori et al. 2011, Agrawal and Goyal 2013, Cheung and Simchi-Levi 2017, Chen et al. 2018a). That is, the benchmark is the solution optimizing the present period. This is acceptable in our application since the dependency in our setting is weak. This is because, for an individual customer, the chance of having another purchase during the considered time horizon is usually small.*

Since parameters θ are unknown at the beginning of the planning horizon, we need to design an algorithm which learns the parameters on the fly and simultaneously maximize the total revenue. The objective to maximize in this chapter is the expected cumulative revenue

$$J^\pi(T) := \mathbb{E}\left[\sum_{t=1}^{T} r(S_t, x_t)\right],$$

where π denotes the algorithm, and S_t is the chosen assortment at time t. The algorithm π has to be non-anticipative in that S_t depends only on the history $\mathcal{F}_{t-1}$ which is the σ-algebra generated by all random variables (e.g., chosen assortment, customer's choice, personal data) until the end of period $t-1$.

The regret. As we discussed earlier, the firm's objective is to design an algorithm which maximizes the expected cumulative revenue $J^\pi(T)$. This objective is equivalent to minimizing the so-called *regret*, which is a commonly used metric to evaluate an online learning algorithm (see e.g., Bubeck et al. 2012), defined as

$$R^\pi(T) := J^*(T) - J^\pi(T) = \mathbb{E}\left[\sum_{t=1}^{T} \big(r(S_t^*, x_t) - r(S_t, x_t)\big)\right].$$

Briefly, the regret $R^\pi(T)$ is the cumulative revenue loss of algorithm π compared with the clairvoyant solution, and our goal is to design a learning algorithm whose regret rate is as low as possible.

3.3 Learning Algorithms and Theoretical Performance

In this section, we present two learning algorithms for the online personalized assortment optimization problem and their regrets. Specifically, we discuss the P-UCB algorithm in Section 3.3.1, and its theoretical performance analysis in Section 3.3.2. In Section 3.3.3, we design a modified version of the P-UCB algorithm, OLP-UCB, which is computationally much more efficient but has similar theoretical performance as P-UCB.

3.3.1 P-UCB algorithm

The main algorithm, which we refer to as Personalized Upper-Confidence-Bound algorithm, or P-UCB algorithm for short, makes dynamic personalized assortment decision in each period t for a customer with personal data x_t. The algorithm for each iteration consists of two steps: 1) Parameter estimation, and 2) Optimization with personalized upper-confidence bound (UCB). Refer to Figure 3.1 for a graphical representation. The detailed algorithm P-UCB is presented below.

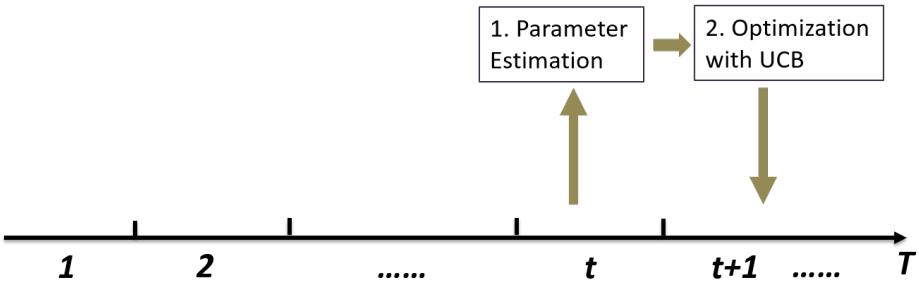


Figure 3.1: The flowchart of algorithm P-UCB

Algorithm overview. In the following, we elaborate on the details of the P-UCB algorithm. Recall that at the start of each iteration $t = 1, \ldots, T$, the decision maker is presented a personal information vector x_t.

Parameter estimation. Using the personal information, the first step of the algorithm estimates $\hat{\theta}_t \in \mathbb{R}^{DN}$, where $\hat{\theta}'_t = (\hat{\theta}'_{1,t}, \ldots, \hat{\theta}'_{N,t})$, using maximum likelihood method based on historical data $\{(S_s, i_s, x_s) : s = 1, \ldots, t-1\}$, where $i_s \in S_s \cup \{0\}$ is the realized customer purchasing decision at period s. More precisely, we have

$$\hat{\theta}_t \in \arg\max_{\theta \in \Theta} \mathcal{L}(\phi | \mathcal{F}_{t-1}), \tag{3.3}$$

Algorithm 3 The P-UCB Algorithm

Require: Confidence bound parameter α.

1: **Step 0. Initialization.** Choose assortment $S_1 \in \mathcal{S}$ randomly and offer to customer at time $t = 1$. Go to Step 1 with time $t = 2$.

2: **Step 1. Parameter Estimation.** Compute the maximum likelihood estimator

$$\hat{\theta}_t \in \arg\max_{\phi \in \Theta} \mathcal{L}(\phi|\mathcal{F}_{t-1}),$$

where $\mathcal{L}(\phi|\mathcal{F}_{t-1})$ is defined in (3.4).

3: **Step 2. Assortment Selection.** Select the assortment S_t according to

$$S_t \in \arg\max_{S \in \mathcal{S}} \hat{r}_t(S, x_t, \hat{\theta}_t),$$

where $\hat{r}_t(S, x_t, \hat{\theta}_t)$ is defined in (3.5).

Let $t = t + 1$ and go to Step 1. The algorithm stops when the end of time horizon is reached.

where $\mathcal{L}(\phi|\mathcal{F}_{t-1})$ is the likelihood function until the end of period $t-1$, given by

$$\mathcal{L}(\phi|\mathcal{F}_{t-1}) := \prod_{s=1}^{t-1} \left(\frac{1}{1 + \sum_{j \in S_s} e^{x_t' \phi_j}} \right)^{Y_{0,s}} \prod_{i \in S_s} \left(\frac{e^{x_t' \phi_i}}{1 + \sum_{j \in S_s} e^{x_t' \phi_j}} \right)^{Y_{i,s}}, \tag{3.4}$$

and $Y_{i,s} \in \{0, 1\}$ represents whether or not the customer purchased product i at time s, i.e., $Y_{i,s} = 1$ if $i_s = i$ and $Y_{i,s} = 0$ otherwise. In the proof of Lemma 3.7.1, we will show that the optimization problem (3.3) is a convex optimization problem hence it can be easily solved. For estimation accuracy, it will be shown that, given sufficient data samples, $\hat{\theta}_t$ is quite close to the true parameter θ, implying that the estimated utility $x_t' \hat{\theta}_{i,t}$ is accurate (close to $x_t' \theta_i$) as well.

Assortment optimization with personalized UCB. The second step of the algorithm is to select the personalized assortment S_t for customer t. Our approach to select S_t is by optimizing a proxy objective function constructed using the personalized upper-confidence bound. That is,

$$S_t \in \arg\max_{S \in \mathcal{S}} \hat{r}_t(S, x_t, \hat{\theta}_t),$$

where

$$\begin{aligned} \hat{r}_t(S, x_t, \hat{\theta}_t) &:= \frac{\sum_{i \in S} p_i \hat{v}_{i,t}(x_t, \hat{\theta}_{i,t})}{1 + \sum_{i \in S} \hat{v}_{i,t}(x_t, \hat{\theta}_{i,t})}, \quad \text{and} \\ \hat{v}_{i,t}(x_t, \hat{\theta}_{i,t}) &:= e^{x_t' \hat{\theta}_{i,t}} + \alpha ||x_t||_{\bar{V}_{i,t}^{-1}}. \end{aligned} \tag{3.5}$$

In the equations above, $\hat{v}_{i,t}(x_t, \hat{\theta}_{i,t})$ is the estimated utility of product i at time t with personalized upper-confidence bound $\alpha||x_t||_{V_{i,t}^{-1}}$, α is the upper-confidence parameter to be specified later, and

$$\begin{aligned} \bar{V}_{i,t} &:= I + V_{i,t} \in \mathbb{R}^{D\times D}, \\ V_{i,t} &:= \sum_{s\in\mathcal{T}_i(t)} x_s' x_s/|S_s| \in \mathbb{R}^{D\times D}, \end{aligned}$$

where $\mathcal{T}_i(t) := \{s < t \ : i \in S_s\}$ is the set of time periods before t that i has been selected in the assortment. Note that $\bar{V}_{i,t}$ can be interpreted as the empirical Fisher's information matrix of product i.

Recall that $||x_t||_{\bar{V}_{i,t}^{-1}}$ is the induced norm by the positive semidefinite matrix $\bar{V}_{i,t}^{-1}$, i.e.,

$$||x_t||_{\bar{V}_{i,t}^{-1}} := \sqrt{x_t' \bar{V}_{i,t}^{-1} x_t}.$$

Intuitively, when product i has been tested only for small number of periods, $\alpha||x_t||_{\bar{V}_{i,t}^{-1}}$ is relatively large (since the minimum eigenvalue $\lambda_{\min}(\bar{V}_{i,t})$ of $\bar{V}_{i,t}$ is small), which suggests to include i in S_t for the purpose of exploration. On the other hand, when sufficient data have been collected for all products $i \in \mathcal{N}$, $\alpha||x_t||_{\bar{V}_{i,t}^{-1}}$ will be small for all i and $\hat{r}_t(S, x_t, \hat{\theta}_t)$ will be close to the real objective function; as a result, $r(S_t, x_t)$ will also be very close to $r(S_t^*, x_t)$. Therefore, in Step 2 of the algorithm, the objective function $\hat{r}_t(S, x_t, \hat{\theta}_t)$ with upper-confidence bound balances the exploration and exploitation by making use of the personalized upper-confidence bound.

Remark 3.3.1 *The assortment selection of S_t in each iteration t of our algorithm is computed using exact and efficient method, see e.g., Rusmevichientong et al. (2010). This is in contrast with Chen et al. (2018a), in which the authors apply an approximation algorithm to compute S_t. This attributes to the different problem formulations for optimization. In our formulation, the optimization problem is (3.5), which is a typical MNL assortment optimization problem with given $\hat{v}_{i,t}$, thus exact (and known) computation method is readily applied; while in Chen et al. (2018a), the authors formulate the assortment optimization problem using an assortment-dependent UCB term, which leads to a complex combinatorial optimization problem destroying the MNL assortment optimization structure, hence they resort to an approximation method to solve it.*

3.3.2 Theoretical performance of P-UCB

The following result presents the theoretical performance of the P-UCB algorithm in terms of regret.

Theorem 3.3.1 *Let $\alpha = cK\sqrt{DN\log(NT/\delta)}$ for any $\delta > 0$, where c is some positive constant, then with probability at least $1-\delta$, the regret of P-UCB algorithm satisfies*

$$\sum_{t=1}^{T}(r(S_t^*, x_t) - r(S_t, x_t)) = O(\alpha\sqrt{DNT\log T}).$$

Taking $\delta = 1/T$, then the expected regret can be written compactly as

$$\mathbb{E}\left[\sum_{t=1}^{T}(r(S_t^*, x_t) - r(S_t, x_t))\right] = \tilde{O}(DNK\sqrt{T}).$$

We offer an explanation on the intuition behind the regret of the algorithm. As we discussed earlier, an important step to bound the regret is to have an accurate estimation of $x_t'\theta_i$. By Cauchy-Schwarz inequality, we have

$$|x_t'\theta_i - x_t'\hat{\theta}_{i,t}| \leq ||\theta_i - \hat{\theta}_{i,t}||_{\bar{V}_{i,t}}||x_t||_{\bar{V}_{i,t}^{-1}}.$$

We call $||\theta_i - \hat{\theta}_{i,t}||_{\bar{V}_{i,t}}$ the confidence bound of product i, and finding its upper bound is crucial for the final regret analysis. It will be shown that this confidence bound is at most $O(K\sqrt{DN\log(NT/\delta)})$ with high probability (i.e., at least $1-\delta$), which ensures the effectiveness of Step 1. Step 2 of the algorithm, as mentioned above, is mainly to balance the exploration and exploitation and bound the regret by the upper-confidence bound. More specifically, for each product i, it will be shown that with high probability,

$$|e^{x_t'\theta_i} - e^{x_t'\hat{\theta}_{i,t}}| \leq \alpha||x_t||_{\bar{V}_{i,t}^{-1}}, \tag{3.6}$$

where

$$\alpha = O\big(K\sqrt{DN\log(NT/\delta)}\big).$$

This means that the utility of each product $e^{x_t'\theta_i}$ is bounded above by

$$\hat{v}_{i,t}(x_t, \hat{\theta}_{i,t}) = e^{x_t'\hat{\theta}_{i,t}} + \alpha||x_t||_{V_{i,t}^{-1}},$$

which is exactly the utility function (3.5) in objective function of Step 2. This result allows

us to show that, when inequality (3.6) is satisfied, the regret in period t satisfies

$$r(S_t^*, x_t, \theta) - r(S_t, x_t, \theta) = O\left(\frac{\alpha}{|S_t|} \sum_{i \in S_t} ||x_t||_{\bar{V}_{i,t}^{-1}}\right).$$

Then, summing over t gives the final regret upper bound.

Lower bound on regret. An immediate question is, what is the lower bound for the regret of our online personalized assortment optimization problem? We note that Chen et al. (2018a) derive a lower bound for a similar problem with different formulation. They formulate the information for each product $i \in \mathcal{N}$ as a D' dimensional vector with all products sharing a common parameter vector. To derive their lower bound, Chen et al. (2018a) assume $N = \Omega(2^{D'})$, which, translating to our problem, would require $N = \Omega(2^{ND})$. This is clearly not satisfied in our setting when N is large. In contrast to the lower bound in Chen et al. (2018a), we have the following lower bound for our problem.

Theorem 3.3.2 *Suppose $D \geq 4$, $N \geq DK$, $T \geq ND \max\{1, 1/R^2\}/144$, for each algorithm π, there exists an instance of the personalized assortment selection problem, such that*

$$R^{\pi}(T) \geq C\sqrt{DNT}/K,$$

where C is a universal constant.

This result implies that the regret of P-UCB algorithm is tight with respect to the time horizon T. But there is some gap in terms of N, K, D (that are typically small compared with T in real applications). We mark this gap as an opportunity for future research.

3.3.3 OLP-UCB: A faster algorithm with online Newton step

In each iteration t of the P-UCB algorithm, the maximum likelihood estimator $\hat{\theta}_t$ is computed with computational time polynomial in N, D and iteration index t. When t becomes large, the calculation of $\hat{\theta}_t$ using all data prior to time t becomes slow. Indeed, in many applications there can be millions of customers in just a few hours, as seen from the Yahoo! example discussed in the introduction section. To overcome this, we develop another algorithm in which the computational cost in each iteration t is constant (i.e., independent of t), and its regret is comparable to that of P-UCB.

The idea of the new algorithm is borrowed from the so-called online-to-confidence set method introduced in Abbasi-Yadkori et al. (2012). Briefly, in each period t, the algorithm calculates an estimated parameter $\bar{\theta}_t$ using an online learning algorithm (in this chapter, we

use an online Newton step). Using parameters $\bar{\theta}_t$, we construct another estimator $\check{\theta}_t$, which will be shown to satisfy that, for any $\delta > 0$ and $i \in \mathcal{N}$,

$$||\check{\theta}_{i,t} - \theta_i||_{\bar{V}_{i,t}} = O\big(K^{3/2}\sqrt{ND\log(T/\delta)}\big)$$

holds with probability at least $1 - \delta$. This allows us to use online learning algorithm to create a small confidence bound for the parameter (this is precisely the reason that such a method is referred to as online-to-confidence set method). Since this algorithm is based on applying an Online (OL) Newton step to the P-UCB algorithm, we call it OLP-UCB algorithm.

Compared with the P-UCB algorithm, the new algorithm OLP-UCB is different only in the parameter estimation step. That is, in Step 1, the estimated parameter is calculated based on a sequence of outputs from an online Newton step.

Online Newton (OL-NEW) step. To motivate, we note that the maximum likelihood estimator $\hat{\theta}_t$ is obtained by minimizing the negative log-likelihood function

$$\begin{aligned} &\min \sum_{s=1}^{t-1} l_s(\phi, Y_s), \quad \text{where} \\ &l_s(\phi, Y_s) = \log\left(1 + \sum_{j \in S_s} e^{x_s'\phi_i}\right) - \sum_{j \in S_s} Y_{i,s} x_s' \phi_i, \end{aligned} \tag{3.7}$$

subjecting to constraint $\phi \in \Theta$, where $Y_s := (Y_{i,s} \ : i \in \mathcal{N})$ represents the customer's choice at time s. The computational cost of optimization problem (3.7) clearly depends on t, which can be expensive as data accumulate, implying that we cannot make quick decisions when t is large.

To accelerate the computation, we note that optimization problem (3.7) is amenable for online convex optimization (see e.g., Hazan et al. 2016). That is, every time when the new data $\{x_t, S_t, Y_t\}$ arrives, we update the estimated parameter $\bar{\theta}_{t+1}$ using the parameter $\bar{\theta}_t$ in the previous iteration and the new available data. There are many online learning algorithms in the literature that can be adopted to solve this problem, among which we choose the so-called online Newton step as an illustrative example. To update the parameter using online Newton step, we first define

$$x_{i,s} := e_i \otimes x_s \in \mathbb{R}^{ND}, \tag{3.8}$$

where $\otimes$ is the Kronecker product and $e_i \in \mathbb{R}^N$ is the unit vector with the entry at location

i being 1 and others 0. Also, define

$$\begin{aligned} \bar{V}_t :=& I + V_t \in \mathbb{R}^{ND\times ND}, \\ V_t :=& \sum_{s=1}^{t-1}\sum_{i\in S_s} x_{i,s}x_{i,s'}/|S_s| \in \mathbb{R}^{ND\times ND}. \end{aligned} \tag{3.9}$$

From these definitions, we can see that $\bar{V}_t$ is a matrix with blocks $\bar{V}_{i,t}$ for $i \in \mathcal{N}$ on its diagonal. Then, the online Newton step updates the parameter recursively by

$$\bar{\theta}^0_{i,t+1} = \bar{\theta}_{i,t} - \frac{\overline{\kappa}^2 K + 2\overline{\kappa} + 1}{\underline{\kappa}} \bar{V}^{-1}_{t+1} \sum_{i\in S_t} \partial_i l_t(\bar{z}_t, Y_t) x_{i,t}, \tag{3.10}$$

where $\bar{z}'_t = (\bar{z}_{i_{1,t},t}, \ldots, \bar{z}_{i_{|S_t|,t},t})$ with $\bar{z}_{i,t} = x'_t\bar{\theta}_{i,t}$ and $i_{1,t} < i_{2,t} < \cdots < i_{|S_t|,t}$ are the indices of products in S_t. With a slight abuse of notation $l_t(\bar{z}_t, Y_t) = l_t(\bar{\theta}_t, Y_t)$ is also a function of $\bar{z}_t$, and we let $\partial_i l_t(\bar{z}_t, Y_t)$ denote the partial derivative of l_t with respect to $\bar{z}_{i,t}$. Clearly, after this update $\bar{\theta}^0_{t+1}$ may be out of range, so a typical approach is to project it onto the feasible set Θ by

$$\bar{\theta}_{t+1} = \arg\min_{\phi\in\Theta} ||\phi - \bar{\theta}^0_{t+1}||_{\bar{V}_{t+1}}, \tag{3.11}$$

where the solution is unique because $\bar{V}_{t+1}$ is positive definite and Θ is convex and compact.

We refer to (3.10) and (3.11) as the Online Newton (OL-NEW) step, which is summarized in the block.

Algorithm 4 The OL-NEW Step

Require: Time period t; Matrix $\bar{V}_{t+1}$; Data in time period t: $\{x_t, S_t, Y_t\}$; Updated parameter $\bar{\theta}_t$ in last period.

1: **Step 1. Update Parameter.** Update $\bar{\theta}^0_{t+1}$ according to

$$\bar{\theta}^0_{t+1} = \bar{\theta}_t - \frac{\overline{\kappa}^2 K + 2\overline{\kappa} + 1}{\underline{\kappa}} \bar{V}^{-1}_{t+1} \sum_{i\in S_t} \partial_i l_t(\bar{z}_t, Y_t) x_{i,t},$$

which is defined in (3.10).

Go to Step 2.

2: **Step 2. Projection and Output.** Project $\bar{\theta}^0_{t+1}$ onto Θ to obtain the final updated parameter $\bar{\theta}_{t+1}$ by

$$\bar{\theta}_{t+1} = \arg\min_{\phi\in\Theta} ||\phi - \bar{\theta}^0_{t+1}||_{\bar{V}_{t+1}}.$$

Then we output $\bar{\theta}_{t+1}$.

Online-to-Confidence Set. In the OL-NEW step, a stream of parameters $\bar{\theta}_1, \bar{\theta}_2, \ldots$ are generated. However, we do not directly take each $\bar{\theta}_t$ as the estimated parameter to obtain a small confidence bound. Instead, we define

$$\boldsymbol{z}'_t := \left(\frac{\bar{z}'_1}{\sqrt{|S_1|}}, \ldots, \frac{\bar{z}'_{t-1}}{\sqrt{|S_{t-1}|}} \right), \tag{3.12}$$

and let $\boldsymbol{X}_t$ be a matrix with $\sum_{s=1}^{t-1} |S_s|$ rows and ND columns such that its rows (from the first to last) are

$$x'_{i_{1,1,1},1}/\sqrt{|S_1|}, \ldots, x'_{i_{|S_1|,1,1},1}/\sqrt{|S_1|}, x'_{i_{1,2,2},2}/\sqrt{|S_2|}, \ldots, x'_{i_{|S_2|,2,2},2}/\sqrt{|S_2|},$$
$$\ldots, x'_{i_{|S_{t-1}|,t-1,t-1},t-1}/\sqrt{|S_{t-1}|}.$$

That is, the $(\sum_{s=1}^{l-1} |S_s| + k)$-th row of the matrix is $x'_{i_{k,l},l}/\sqrt{|S_l|}$. Then, we define

$$\check{\theta}^0_t := \bar{V}_t^{-1} \boldsymbol{X}'_t \boldsymbol{z}_t \in \mathbb{R}^{ND}.$$

Note that the computational cost of $\check{\theta}^0_t$ at each time t is independent of t because

$$\boldsymbol{X}'_t \boldsymbol{z}_t = \boldsymbol{X}'_{t-1} \boldsymbol{z}_{t-1} + \sum_{i \in S_{t-1}} \bar{z}_{i,t-1} x_{i,t-1} / |S_{t-1}|.$$

Hence the additional computation in iteration t is from $\sum_{i \in S_{t-1}} \bar{z}_{i,t-1} x_{i,t-1} / |S_{t-1}|$. Our final estimator is then defined by projection

$$\check{\theta}_t := \arg\min_{\phi \in \Theta} ||\phi - \check{\theta}^0_t||_{\bar{V}_t}.$$

The detailed OLP-UCB algorithm is presented in the block.

The following result presents the theoretical performance of OLP-UCB algorithm, which is similar to that of P-UCB (except the factor $\sqrt{K}$ and some constants).

Theorem 3.3.3 *Let $\alpha = cK^{3/2}\sqrt{DN \log(T/\delta)})$ where c is some positive constant, then with probability at least $1 - \delta$, the regret of algorithm OLP-UCB is*

$$\sum_{t=1}^{T} (r(S^*_t, x_t) - r(S_t, x_t)) = O(\alpha \sqrt{DNT \log T}).$$

Algorithm 5 The OLP-UCB Algorithm

Require: confidence bound α.

1: **Step 0. Initialization.** Choose assortment $S_1 \in \mathcal{S}$ randomly and offer to customer at time $t = 1$. Let $\check{\theta}_1 \in \Theta$ and $\bar{\theta}_1 \in \Theta$ be selected uniformly randomly. Go to Step 1 with time $t = 2$.

2: **Step 1.1. Online Newton step.** Given data in $t-1$: $\bar{V}_t$, $\{x_{t-1}, S_{t-1}, Y_{t-1}\}$, $\bar{\theta}_{t-1}$, apply algorithm OL-NEW and output $\bar{\theta}_t$. Update $\bar{z}_t' = (\bar{z}_{i_1,t}, \ldots, \bar{z}_{i_{|S_s|},t})$ with $\bar{z}_{i,t} = x_t'\bar{\theta}_{i,t}$ and $i_1, \ldots, i_{|S_t|}$ represents the indices of products in S_t.

3: **Step 1.2. Parameter Estimation.** Compute the estimator $\check{\theta}_t^0 := \bar{V}_t^{-1}\boldsymbol{X}_t'\boldsymbol{z}_t$, where $\boldsymbol{z}_t$ is given by (3.12), and $\boldsymbol{X}_t$ is a matrix with $\sum_{s=1}^{t-1}|S_s|$ rows and ND columns such that each row is $x_{i,s}'/\sqrt{|S_s|}$. Then define

$$\check{\theta}_t := \arg\min_{\phi\in\Theta} ||\phi - \check{\theta}_t^0||_{\bar{V}_t}.$$

4: **Step 2. Assortment Selection.** Select assortment S_t according to

$$S_t \in \arg\max_{S\in\mathcal{S}} \hat{r}_t(S, x_t, \check{\theta}_t),$$

where $\hat{r}_t(S, x_t, \check{\theta}_t)$ is defined in (3.5).

Let $t = t + 1$ and go to Step 1. The algorithm stops when the end of time horizon is reached.

Taking $\delta = 1/T$, the expected regret of the algorithm can be written compactly as

$$\mathbb{E}\left[\sum_{t=1}^{T}(r(S_t^*, x_t) - r(S_t, x_t))\right] = \tilde{O}(DNK^{3/2}\sqrt{T}).$$

3.4 Solving High Dimensional Problem via Random Projection

In real world applications of online personalized assortment optimization, customer's personal information data can have high dimension (i.e., D is extremely large). Take online shopping website as an example. The clicking/purchasing history of a customer can be represented by a vector of binary variables, in which each entry corresponds to a product offered on the website. If a customer has clicked/purchased a specific product before (within certain period of time), the corresponding entry of that product is marked as 1; otherwise it is 0. Since online shopping websites, such as Amazon.com and Taobao.com (the largest global C2C shopping website owned by Alibaba Group), contain millions of products, it is conceivable that the vector of historical data has extremely high dimension. This high

dimensionality of the personal information data (in particular, the historical data) leads to at least two issues for the decision maker. First, if demand estimation is applied using generic high dimensional data, the theoretical performance of the learning algorithms (e.g., results in Theorem 3.3.1 and Theorem 3.3.3) is bad because of large D. Second, the computation (even for a simple computation like vector addition or matrix-vector multiplication) using high dimensional data becomes very inefficient.

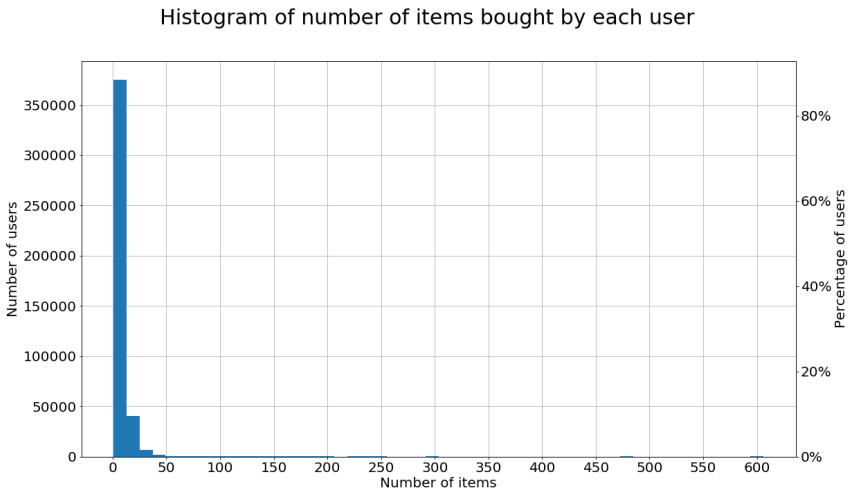
To gain some understanding of the structure of personal information data, we have conducted a data analysis of a real dataset of customers' purchasing records from Alibaba Group[1]. This dataset contains the purchasing records from $424,170$ customers and $372,740$ products with time spanned from May to November in 2014. Although this dataset indicates that customer's personal information data has a dimension of at least $372,740$, most customers purchased very few products. In particular, data show that the most popular product was purchased by 3145 customers ($< 0.8\%$ of all customers), and a majority of the products (99% of the products) were purchased by fewer than 106 customers ($< 0.03\%$ of all customers). Furthermore, the most active shopper purchased 606 products (only $< 0.17\%$ of all products), and more than 99% of the customers purchased fewer than 31 products ($< 0.009\%$ of all products). Refer to Figure 3.2 for a graphic representation of this sparsity structure. We can see that the two histograms are extremely skewed to the scenario that the number of buyers/items is very small, implying that the customer's (historical purchasing) data is sparse.

From the example above, we see that although the data is extremely high dimensional, it is quite sparse. Therefore in this section, we assume high dimensional sparse data, and address the question that whether one can make personalized assortment decision effectively (i.e., to maximize revenue) and efficiently (i.e., with low computational complexity). To this end, we make the following assumption on the problem, which reflects the observation from the real data. Here the notation $x_{t;k}$ represents the k-th entry of vector x_t.

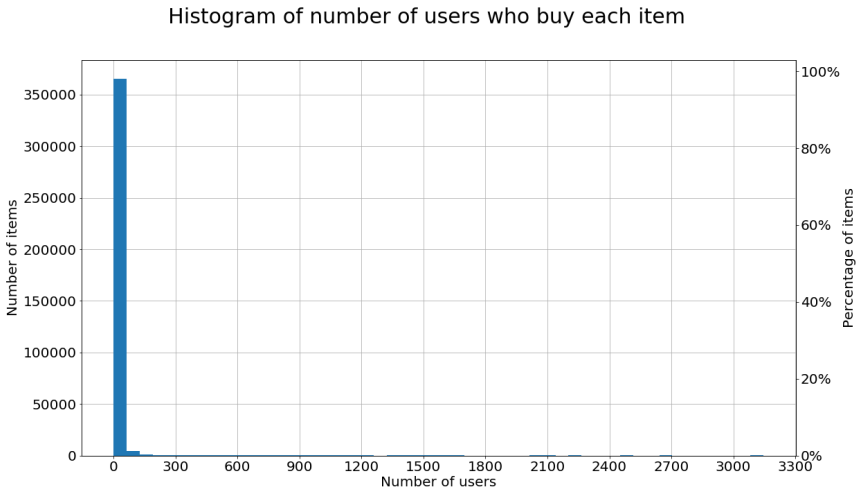
Assumption 1. There exist some nonnegative integers d_0, L such that for any $k > d_0$, the data satisfy $|\{x_{t;k} \neq 0 : t \in \mathcal{T}\}| \leq L$.

This sparsity assumption requires that for all but d_0 features in the vector x_t, at most L customers have nonzero entries. For example, in our data analysis above in Figure 3.2(b), this means that most of the items are purchased/clicked by relatively few users. This assumption is different from sparsity of vectors x_t. It can be argued that, when vectors x_t are sparse, i.e., all but a few x_t for $t \in \mathcal{T}$ are sparse, there exist some d_0 and L, which are small compared with D and T, such that the above assumption is satisfied. To see that, suppose for ease of discussion all x_t are sparse, i.e., each x_t has at most, say, a non-zero

[1]The dataset is publicly available at https://tianchi.aliyun.com/dataset/dataDetail?dataId=47



(a) Number of items each user purchased



(b) Number of users who purchased each item

Figure 3.2: Histogram of number of items each user purchased and number of users who purchased each item

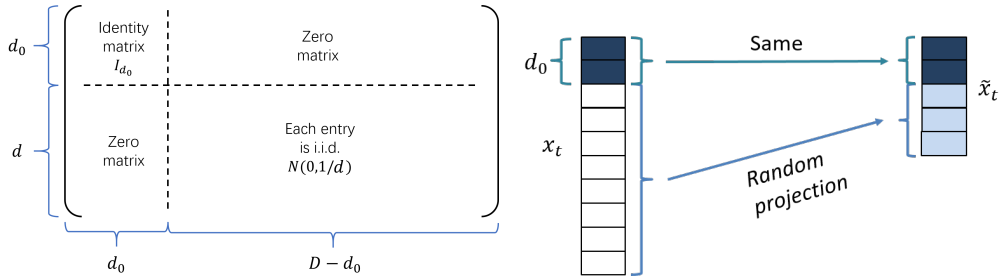
entries, where a is small compared with D and aT/D is small compared with T. When all the non-zeros of x_t are for the same features, then it becomes our model with $d_0 = a$ and $L = 0$; while in the other extreme case that the non-zeros of x_t are evenly allocated to the features, then each column of data matrix (with rows as x_t' for $t \in \mathcal{T}$) will have at most aT/D non-zero elements, which is equivalent to our model with $d_0 = 0$ and $L = aT/D$. Generally, i.e., at most a elements in x_t are nonzero and they form the rows of matrix X, since the total number of nonzero entries for all columns is at most aT, if there are at most d_0 columns with more than L nonzero entries, then we have $d_0 L \leq aT$. Thus, if we let $d_0 = \lceil D/m \rceil$ for some positive number m, then we have $L \leq \lceil maT/D \rceil$.

Our approach and analysis differ from the existing literature on high dimensional estimation using LASSO (see e.g., Tibshirani 1996, Bastani and Bayati 2015, Ban and Keskin 2017). First, LASSO theory typically assumes that the underlying parameter θ is sparse, which cannot be verified in reality as they are not known *a priori*; we assume sparsity of customers' data which is observable to the firm. Second, LASSO is computationally inefficient, hence failing to address the challenge of high computational cost. In this chapter, we apply a dimensionality reduction approach known as *random projection* (see e.g., Kaban 2015 for an introduction) to tackle this problem. Briefly, we project the high dimensional data into a low dimensional space such that all computations (e.g., parameter estimation) are performed in the low dimensional space. This dimension reduction obviously accelerates the computation, especially as the projection of data can be performed before the algorithm starts because the firm has access to the database. Nonetheless, we will show that the error caused by random projection is not significant when the data is sparse, and that the performance of the new algorithm is very good.

It is worth noting that there are other methods of dimension reduction. For instance, Chapelle and Li (2011) use principal component analysis (PCA, see e.g., Jolliffe 2011 for an overview) to preprocess the high dimensional data into low dimensional vectors, and then apply a common learning approach to the low dimensional data. However, to the best of our knowledge, there exist no research in the literature on the effect of these dimension reduction methods on the regret of online learning algorithms; thus this study will be the first to fill this gap.

Note that although x_t is assumed to be sparse, the indices of nonzero entries of each x_t are usually different. Thus the unknown vector θ has to be learned using all the cumulative data x_t. By Assumption 1, there is a dense portion in each x_t, so a simple dimension reduction method is to only keep the dense portion and ignore the sparse part. In our numerical study, we will also test the performance of this simple heuristic based on a real dataset, in Section 3.5.3.

Random projection. Let $M \in \mathbb{R}^{(d_0+d)\times D}$ be a random matrix whose entry at row k and column l is $M_{k,l}$, where $d \ll D$ is a positive integer to be specified later. Let $M_{k,k} = 1$ and $M_{k,l} = 0$ for $k \neq l$ and k and l no more than d_0, and $M_{k,l} = 0$ for $k \leq d_0, l > d_0$ or $k > d_0, l \leq d_0$, For $k, l > d_0$, $M_{k,l}$ are i.i.d. sub-Gaussian random variables with mean 0 and variance $1/d$. See Figure 3.3(a) for an illustration. Using random matrix M, we project the high dimensional vector x_t into low dimensional space of dimension $d_0 + d$ by $\tilde{x}_t = Mx_t$. This projection keeps the dense portion (i.e., entries $k \leq d_0$) of each vector x_t unchanged, but projects the high dimensional sparse part to a d-dimensional space. Refer to Figure 3.3 for an illustration of the random matrix and projection.



(a) Block structure of matrix M using Gaussian distribution as an example

(b) How matrix M projects x_t to $\tilde{x}_t$

Figure 3.3: A graph representation of random projection matrix M

With the projected $(d_0 + d)$-dimensional data $\tilde{x}_t$, we apply the OLP-UCB algorithm from the previous section to solve this low-dimensional problem. We refer to this OLP-UCB algorithm based on random projection (RP) as *OLP-UCB-RP* algorithm.

For brevity, we will not repeat the entire algorithm here, but only highlight the differences with the original algorithm. First of all, let $\epsilon \in (0, 1)$ be an input parameter related to the random projection error with projected dimension d. Then define

$$\tilde{\Theta}_i := \{\phi \in \mathbb{R}^{d_0+d} : ||\phi||_2 \leq \sqrt{1+\epsilon}R\}, \qquad \tilde{\Theta} := \bigotimes_{i=1}^{N} \tilde{\Theta}_i.$$

Next, let

$$\bar{W}_{i,t} := (1+\epsilon)I + W_{i,t},$$
$$W_{i,t} := \sum_{s \in \mathcal{T}_i(t)} \tilde{x}_t \tilde{x}_t' / |S_s|$$

be the empirical Fisher's information matrix for all $i \in \mathcal{N}$ in the projected space. Then algorithm OL-NEW and OLP-UCB are applied in the projected space in that all x_t are replaced by projected $\tilde{x}_t$ (and $\boldsymbol{X}_t$ and $\boldsymbol{z}_t$ are changed to their projected version $\tilde{\boldsymbol{X}}_t$ and $\tilde{\boldsymbol{z}}_t$ respectively), $\bar{V}_{i,t}$ replaced with $\bar{W}_{i,t}$ for all $i \in \mathcal{N}$ and $t \in \mathcal{T}$, and $\bar{\theta}_t$ and $\check{\theta}_t$ are computed in the low dimensional feasible set $\tilde{\Theta}$.

Before presenting the theoretical result on the performance of OLP-UCB-RP algorithm, we make the following technical assumption. Let $x_{t;>d_0}$ denote the sparse portion of x_t, i.e., $x_{t;>d_0} = (x_{t;k} : \ k > d_0)$.

Assumption 2. There exists a constant $\gamma > 0$ such that $|x'_{t;>d_0}\theta_{i;>d_0}| \geq \gamma ||x_{t;>d_0}||_2 ||\theta_{i;>d_0}||_2$ for all $i \in \mathcal{N}$ and $t \in \mathcal{T}$.

Geometrically, this assumption requires that the cosine of the angle between $x_{t;>d_0}$ and $\theta_{i;>d_0}$ be at least γ, so the two vectors are not orthogonal to each other. In our problem, this

assumption implies that the information corresponding to the sparse portion of x_t is useful for all products $i \in \mathcal{N}$ and time $t \in \mathcal{T}$. If this assumption is not satisfied, e.g., imagine that for most $t \in \mathcal{T}$ with $x_{t;>d_0} \neq 0$, we have $|x'_{t;>d_0}\theta_{i;>d_0}| = 0$, then the sparse portion $x_{t;>d_0}$ can be ignored for i and the parameter θ_i is reduced to $\theta_{i;\leq d_0}$. We remark that, it is possible to relax Assumption 2 such that for each $i \in \mathcal{N}$, some x_t violate this assumption, and a modified performance bound can be obtained. However, for ease of presentation, we shall assume Assumption 2 holds for all i and t. In Section 3.5.3, we will test the algorithm using a real dataset when the entire sparse portion is dropped.

The theoretical performance of the algorithm using random projection is given in the following result.

Theorem 3.4.1 *Let $\alpha = cK^{3/2}\sqrt{N}(\epsilon/\gamma\sqrt{L} + \sqrt{(d_0+d)\log(T/\delta)})$ where c is some positive constant, and $d \geq 8\log(8TN/\delta)/\epsilon^2$, then with probability at least $1-5\delta$, the regret of algorithm OLP-UCB-RP is*

$$\sum_{t=1}^{T}(r(S_t^*, x_t) - r(S_t, x_t)) = O(\epsilon/\gamma\sqrt{LNT} + \alpha\sqrt{(d+d_0)NT\log T}).$$

In a more compact form, by letting $\delta = 1/T$, we have

$$\mathbb{E}\left[\sum_{t=1}^{T}(r(S_t^*, x_t) - r(S_t, x_t))\right] = \tilde{O}(\epsilon/\gamma NK^{3/2}\sqrt{(d_0+d)LT} + (d_0+d)NK^{3/2}\sqrt{T}).$$

By Theorem 3.4.1, the regret of OLP-UCB-RP algorithm has two parts: the first part $O(\epsilon/\gamma\sqrt{LNT})$ is the regret from projection, while the second part $O(\alpha\sqrt{(d+d_0)NT\log T})$ is regret from the projected low dimensional space. For convenience we shall call them projection regret and projected space regret, respectively. We shall prove Theorem 3.4.1 by separately analyzing these two regrets.

3.5 Numerical Experiments

In this section, we present the results for several numerical experiments on our algorithms. To demonstrate their performances, we use the MNL-Bandit algorithm proposed in Agrawal et al. (2017a) as a benchmark.

This section consists of three parts. In Section 3.5.1, an illustrative synthetic dataset is used to simulate the customers' selection behavior, and all algorithms are tested according to this simulation. Besides showing the effectiveness of P-UCB and OLP-UCB algorithms,

another purpose of this experiment is to show that OLP-UCB, that has slightly worse theoretical performance but runs much faster, performs nearly as good as P-UCB numerically. In Section 3.5.2, we test all algorithms using a real dataset provided by Yahoo!. Because of the similarity in numerical performance between P-UCB and OLP-UCB algorithms as shown in Section 3.5.1, we will only test OLP-UCB on the real data due to its efficiency in computation (as the real dataset is quite large). A method developed by Li et al. (2011) will be used to estimate the unbiased performance of an online learning algorithm when applied in real life setting. Finally in Section 3.5.3, we test the performance of OLP-UCB-RP algorithm on high dimensional personal data.

3.5.1 Numerical experiments with synthetic data

The synthetic dataset is generated as follows. Let the length of time horizon $T = 10,000$, and the original dimension of data $D = 6$. Each x_t is generated randomly such that $x_{t;1} = 1$ for all $t \in \mathcal{T}$, and the rest of $x_{t;k}$ are drawn uniformly from $[-1, 1]$. For the products, we let $N = 10$ and $K = 4$ with prices $p_i \in [0, 1]$ chosen uniformly for all $i \in \mathcal{N}$. Each $\theta_{i;k}$ is drawn uniformly from $[0, 1]$. After generating all the data, we normalize all x_t to $||x_t||_2 \leq 1$ and all θ_i to $||\theta_i||_2 \leq R = 10$.

We run each algorithm for 30 experiments, and take their average as the output. Note that both P-UCB and OLP-UCB need a tuning parameter c (or equivalently, α). Parameter tuning for machine learning algorithms is nontrivial and extensive research has been conducted on this topic (for example, Bayesian optimization has been used for parameter tuning, see e.g., Snoek et al. 2012, Frazier and Wang 2016). In this chapter, we choose the tuning parameter c such that $\alpha = 0.5$ ad-hoc for both algorithms. Note that except the regret, we also use another metric called *percentage revenue loss* for each algorithm π defined as

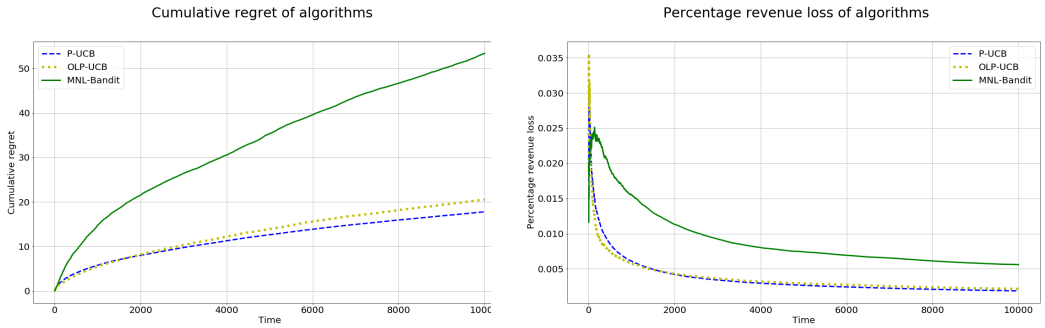
$$L^{\pi}(T) := \frac{R^{\pi}(T)}{J^{*}(T)} \times 100\%,$$

which is the percentage of revenue loss compared with the clairvoyant optimal solution.

	P-UCB	OLP-UCB	MNL-Bandit
Mean	0.19%	0.22%	0.56%
Standard deviation	0.09%	0.03%	0.04%

Table 3.1: Mean and standard deviation of percentage revenue loss for all algorithms

The results are shown in Figure 3.4 (the blue dashed line is for P-UCB; the yellow dotted line is for OLP-UCB; the green solid line is for MNL-Bandit), with the mean and standard deviation of the percentage revenue loss for each algorithm given in Table 3.1.



(a) Cumulative regrets

(b) Percentage revenue loss

Figure 3.4: Cumulative regrets and percentage revenue loss for different algorithms.

From these results, we can see that both P-UCB and OLP-UCB algorithms, whose performances are quite close to each other, significantly outperform the benchmark algorithm (MNL-Bandit) that ignores customers' data. The standard deviation of percentage revenue loss for our algorithms show that they are also very robust. In terms of computational time, when $t = 10,000$, the average computational time (of one time period) for OLP-UCB is 0.4105 seconds (compared with 0.0165 seconds when $t = 100$) on a personal laptop (Processor: Intel(R) Core(TM) i7-4600M CPU @ 2.90 GHz 2.90 GHz; RAM: 8GB; System type: 64-bit Operating System, x64-based processor), while for OLP-UCB, it is constantly around 0.0021 seconds independently of t. Because P-UCB and OLP-UCB have performances very close to each other but OLP-UCB is computationally much faster, we will focus on OLP-UCB in the next subsection on our experimentation with real data.

3.5.2 Numerical experiments with real data

How to obtain the true (unbiased) performance of a new algorithm in a real setting without implementing the algorithm on site? This is a difficult but important question. Fortunately, Li et al. (2010) developed a method which achieves this using real historical data. But for their method to work, the real data has to satisfy a strong "uniformly random selection" condition. In this subsection, we use a dataset from Yahoo! that satisfies the required condition; therefore it can be used to demonstrate the performance of our algorithms in the real life setting.

The method is described as follows. Suppose we have a dataset of $\{(x_t, \hat{S}_t, \hat{i}_t)|\ t = 1, \ldots, T\}$ where x_t is the context, $\hat{S}_t$ is the offered assortment, and $\hat{i}_t$ is the customer's choice. The selling price of product i is p_i. We aim to study the performance of a new learning algorithm, say OLP-UCB, using this dataset. For each $t = 1, 2, \ldots,$ given x_t in period t as the input, suppose our learning algorithm suggests decision S_t. If $S_t \neq \hat{S}_t$, then

this data sample is ignored and we move to $t+1$; otherwise, our algorithm collects revenue $p_{\hat{i}_t}$ from customer choosing $\hat{i}_t$, and this data sample $(x_t, \hat{S}_t, \hat{i}_t)$ is included as the outcome of the learning algorithm. Let $N(T)$ denote the total number of data samples matched by the algorithm, then the revenue accumulated from these matched samples represents the revenue of the learning algorithm from $N(T)$ periods. Li et al. (2011) proved that, if for each x_t the assortment $\hat{S}_t$ in the dataset was selected *uniformly randomly* from all possible choices, then the revenue generated according to this process replicates the revenue of our algorithm in the real setting with no bias. We point out that this method has been applied in studying the performance of bandit algorithms using real data in, e.g., Li et al. (2010), Strehl et al. (2010), Li et al. (2011), May et al. (2012).

Description of the dataset. Yahoo! provided us with such a dataset which is called *Yahoo! Front Page Today Module User Click Log Dataset, version 2.0*. This dataset contains users' clicking reactions to recommended news articles on the front page of Yahoo!'s website from 10/2/2011 to 10/16/2011. In this dataset, each row is a user visit, e.g., in the following row "1317513291 id-560620 0 | user 1 9 11 13 23 16 18 17 19 15 43 14 39 30 66 50 27 104 20 | id-552077 | id-555224 | $\cdots$ | id-565822", each entry has the following meaning:

- Time-stamp: 1317513291
- Displayed article ID: id-560620
- User action: 0 (representing that user did not click article id-560620, and 1 for click)
- The string "user" indicates the start of user's feature x_t
- User features: each user has a feature of 136 binary variables; each number after the string "user" represents the index of nonzero entry in the feature
- The remaining | id-552077 | id-555224 | $\cdots$ | id-565822 is the set of article candidates that can be recommended

To assess the performance of our algorithms, we select around $340,000$ rows from the dataset in 10/09/2011, with each row having explicit feature information. We note that for this day, 32 articles (i.e., $N = 32$) are constantly available to offer to users, and in this dataset only one article is recommended (i.e., $K = 1$). As stated in the Yahoo!'s website[2], the recommended "articles were chosen uniformly at random", thus allows us to obtain unbiased evaluation of our algorithms.

[2]https://webscope.sandbox.yahoo.com\catalog.php?datatype=r. Accessed July 17, 2019.

The measure we use to evaluate an learning algorithm is the average clicking through rate (CTR) which is defined as the ratio between the total number of clicks observed by the learning algorithm and the total number of recommendations (excluding those ignored by the process because of unmatched recommendation). As indicated in the previous subsection, we will only test OLP-UCB because of its computational efficiency (especially given that the number of customers in this dataset is large and the feature dimension 136 is moderately high). For parameter tuning, after a few simple tries (i.e., using a small fraction of data for testing different α), we set $\alpha = 0.01$. Besides the benchmark MNL-Bandit algorithm, we also plot the results of uniformly random selection algorithm which is exactly how the item was recommended to each customer in this dataset.

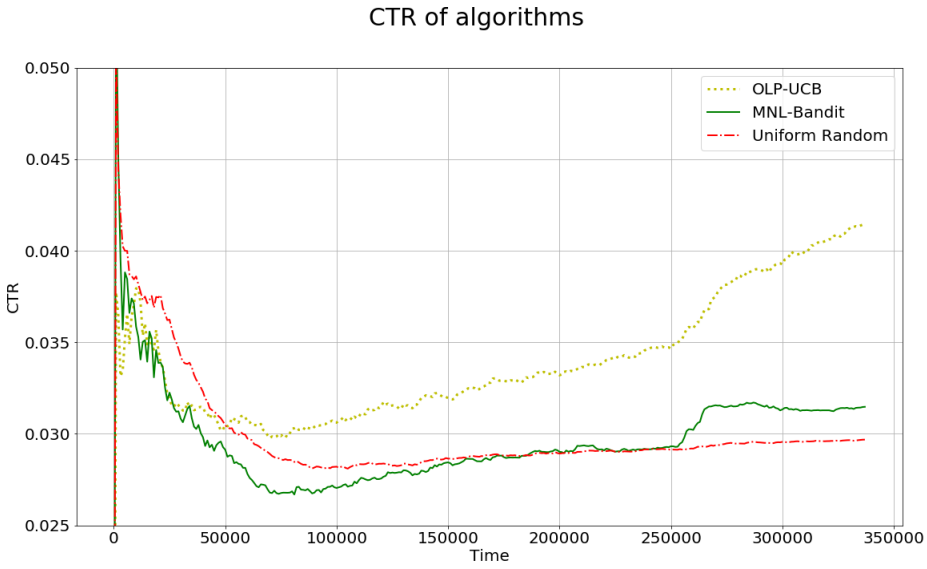


Figure 3.5: CTR for different algorithms.

The results of the CTR for different algorithms are shown in Figure 3.5. The yellow dotted line is for OLP-UCB; the green solid line is for MNL-Bandit; the red dash-dotted line is for uniformly random selection algorithm. Note that the largest number 340,000 is the number of customers (rows) we tested over the several hours of the day. According to the experiment result, the uniformly random selection algorithm (which is what was applied in reality) has an overall CTR equal to 2.97%; MNL-Bandit algorithm slightly improves the result and has overall CTR equal to 3.15% (a 6.06% increase from the CTR of uniformly random selection algorithm); our algorithm OLP-UCB has the best performance with overall CTR equal to 4.15%, which is 39.73% higher than uniformly random selection algorithm and 31.75% higher than MNL-Bandit algorithm.

3.5.3 Numerical experiments for high dimensional data

In this section, we present the numerical results of OLP-UCB-RP algorithm on high dimensional data. For parameter tuning, we simply choose the same c as in the low dimensional case, and let the projected dimension $d = 30$ as an illustrating example. In addition to comparing with the MNL-Bandit algorithm, we also compare the results with OLP-UCB using original data (i.e., personalized high dimensional data before projection).

The synthetic dataset is generated as follows. The length of time horizon is $T = 10,000$, and the original dimension of data $D = 1,001$. For each x_t, let $x_{t;1} = 1$ so the first entry is 1; for the rest of its entries $k > 1$, we let $x_{t;k} = 0$ with probability 0.9 and $x_{t;k} = 1$ with probability 0.1. This construction guarantees the sparsity in that each x_t has roughly 10% of its entries not equal to zero. For all customers $t \in \mathcal{T}$, we assume that there are in general two different groups of customers (with equal probability). Specifically, for the first group, if $k \in \{2, \ldots, 501\}$, each nonzero $x_{t;k} = 1$ with probability 0.9, and $x_{t;k} = -1$ with probability 0.1; if $k \in \{502, \ldots, 1001\}$, each nonzero $x_{t;k} = -1$ with probability 0.9, and $x_{t;k} = 1$ with probability 0.1. For the second group, the nonzero $x_{t;k} \in \{\pm 1\}$ in the reverse manner, i.e., they are equal to 1 with probability 0.1 and -1 with probability 0.9 for $k \in \{2, \ldots, 501\}$ and the opposite for $k \in \{502, \ldots, 1001\}$. This synthetic customers' data can be considered as in the following example. A music/movie recommendation website has two genres of contents, A and B, that are dramatically different, to recommend to customers. The website has access to customers' rating history of similar contents, with 1 representing "like" for that content, -1 representing "dislike", and 0 representing no rating history. In x_t, $x_{t;k}$ for $k \in \{2, \ldots, 501\}$ represent similar contents for genre A, and $k \in \{502, \ldots, 1001\}$ represent similar contents for genre B. Obviously, customers in the first group have more positive ratings for genre A and negative ratings for genre B; customers in the second group have the opposite opinion. Again, we normalize the x_t to $||x_t||_2 \leq 1$.

For the products, we let $N = 10$ and $K = 4$, with price $p_i \in [0, 1]$ chosen uniformly for all $i \in \mathcal{N}$ again as in Section 3.5.1. As in the example above, these 10 products belong to two genres A and B. Suppose products $i = 1, \ldots, 5$ belong to genre A, and products $i = 6, \ldots, 10$ belong to genre B. For a product of genre A, we let $\theta_{i;k}$ be uniformly from $[1, 0]$ for all $k \in \{2, \ldots, 501\}$, and $\theta_{i;k}$ generated uniformly from $[-1, 0]$ for all $k \in \{502, \ldots, 1001\}$. For product i belonging to genre B, θ_i is generated oppositely. Finally, we let $\theta_{i;1}$ be generated from $[-1, 1]$ uniformly for all $i \in \mathcal{N}$. Again, we normalize the θ_i to $||\theta_i||_2 \leq R = 10$.

The numerical results are summarized in Table 3.2. As seen, OLP-UCB-RP achieves an average percentage revenue loss rate of 2.53% at the end of horizon, which is signifi-

	$T = 1000$		$T = 3000$		$T = 6000$		$T = 10000$	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
OLP-UCB-RP	4.38%	0.49%	3.32%	0.46%	2.81%	0.46%	2.53%	0.46%
OLP-UCB	1.9%	0.33%	1.20%	0.17%	1.04%	0.12%	1.02%	0.09%
MNL-Bandit	7.39%	0.27%	7.23%	0.11%	7.20%	0.07%	7.15%	0.04%

Table 3.2: Percentage revenue loss for different algorithms

cantly lower than the 7.15% of the MNL-Bandit algorithm. It can be seen that OLP-UCB, that uses the original high dimensional data, achieves a better percentage revenue loss rate numerically. Indeed, the purpose of OLP-UCB-RP is to yield good performance in a short computational time. The average computational time for each time period t of OLP-UCB-RP is only 0.0027 seconds (on the same personal laptop as in the previous subsection), while for OLP-UCB it is 1.5125 seconds. This implies that OLP-UCB-RP will be a preferred option in real applications when the problem dimension becomes extremely high.

We further test OLP-UCB-RP using the real dataset in Section 3.5.2. In this setting, the dense portion of x_t is picked such that for each entry $x_{t;k}$, at least 40% of the time t have $x_{t;k} \neq 0$; since the dimension of x_t is 136, which is not very high, we choose the projected dimension $d = 10$ as an illustrating example. Besides OLP-UCB-RP, we also test another benchmark named as OLP-UCB-Dense, which is just applying OLP-UCB on the dense portion of x_t (i.e., the sparse portion of each x_t is completely ignored). The reason to add this benchmark is to test whether the information in the sparse portion of x_t can help in decision making, or it introduces too much noise for making better decisions.

The numerical results are depicted in Figure 3.6. Compared with Figure 3.5, Figure 3.6 also includes results for OLP-UCB-RP (blue dashed line) and OLP-UCB-Dense (black solid line with mark). According to these results, the overall CTR of OLP-UCB-RP is 4.15%, which is almost the same as OLP-UCB. Moreover, OLP-UCB-Dense achieves an overall CTR 3.88%, which is worse than that of OLP-UCB-RP and OLP-UCB. This implies that the sparse portion of the data does help with maximizing the overall CTR. Therefore, we recommend practitioners to include sparse data using dimension reduction method instead of ignoring them.

3.6 Sketches of Proofs

In this section we only outline the main steps in the proofs of Theorem 3.3.1 and Theorem 3.4.1, while leaving the technical details in a series of lemmas, together with the proof

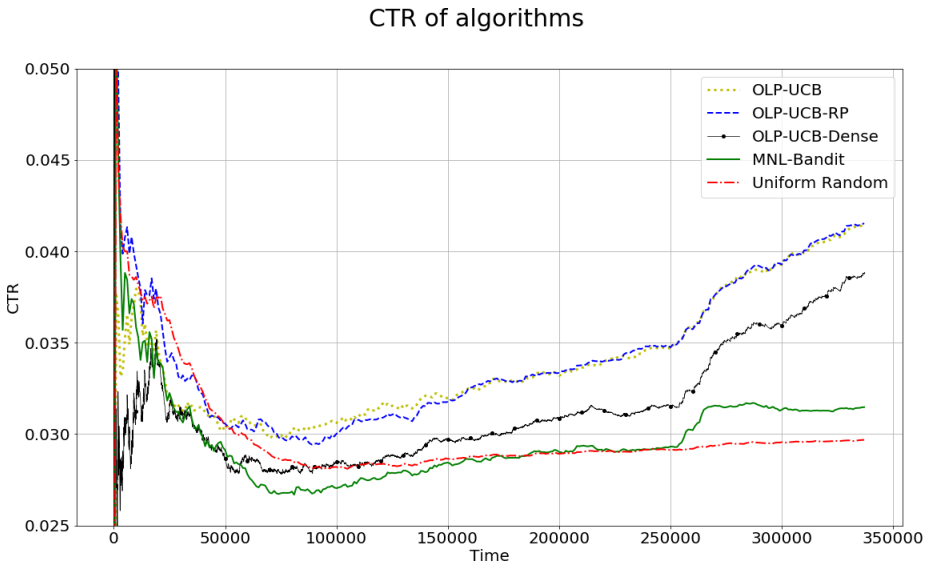


Figure 3.6: CTR for different algorithms.

of lower bound result, i.e., Theorem 3.3.2, in the section of proofs. Since the proof of Theorem 3.3.3 is similar to that of Theorem 3.3.1 (with the only difference being the upper bound for $||\check{\theta}_{i,t} - \theta_i||_{\bar{V}_{i,t}}$ which is established in Lemma 3.7.5 in the section of proofs), we will omit it for brevity.

3.6.1 Outline of proof of Theorem 3.3.1

The cumulative regret can be expressed as

$$R(T) = \sum_{t=1}^{T} \big(r(S_t^*, x_t, \theta) - r(S_t, x_t, \theta)\big).$$

We define the event

$$\mathcal{E}_0 := \Big\{||\theta_i - \hat{\theta}_{i,t}||_{\bar{V}_{i,t}} \leq k_1 K \sqrt{DN \log(NT/\delta)}, \forall i \in \mathcal{N}, t \in \mathcal{T}\Big\}$$

for some constant k_1. This is the event that the MLE is close to the true parameter estimated, and in Lemma 3.7.1 in the section of proofs, we will show that it occurs with high probability, or $\mathbb{P}(\mathcal{E}_0) \geq 1-\delta$ for any $\delta > 0$. This allows us to construct an upper confidence bound which is crucial for regret analysis.

Conditioning on $\mathcal{E}_0$, we have that for all $i \in \mathcal{N}$ and $t \in \mathcal{T}$,

$$|e^{x_t'\hat{\theta}_{i,t}} - e^{x_t'\theta_i}| \leq \overline{\kappa}|x_t'\hat{\theta}_{i,t} - x_t'\theta_i| \leq \overline{\kappa}||\theta_i - \hat{\theta}_{i,t}||_{\bar{V}_{i,t}}||x_t||_{\bar{V}_{i,t}}^{-1} \leq \alpha||x_t||_{\bar{V}_{i,t}}^{-1},$$

where the second inequality follows from Cauchy-Schwarz and the last inequality is from event $\mathcal{E}_0$ with our choice of α (by taking $c \geq \overline{\kappa}k_1$). As a result, $\hat{v}_{i,t}(x_t, \hat{\theta}_{i,t}) \geq \exp(x_t'\theta_i)$ for all $i \in \mathcal{N}, t \in \mathcal{T}$. This implies that

$$\begin{aligned}
r(S_t^*, x_t, \theta) - r(S_t, x_t, \theta) =& r(S_t^*, x_t, \theta) - \hat{r}_t(S_t, x_t, \hat{\theta}_t) + \hat{r}_t(S_t, x_t, \hat{\theta}_t) - r(S_t, x_t, \theta) \\
\leq& \hat{r}_t(S_t, x_t, \hat{\theta}_t) - r(S_t, x_t, \theta) \\
=& \frac{\sum_{i\in S_t} p_i \hat{v}_{i,t}(x_t, \hat{\theta}_{i,t})}{1 + \sum_{i\in S_t} \hat{v}_{i,t}(x_t, \hat{\theta}_{i,t})} - \frac{\sum_{i\in S_t} p_i e^{x_t'\theta_i}}{1 + \sum_{i\in S_t} e^{x_t'\theta_i}} \\
\leq& \frac{\sum_{i\in S_t} p_i (\hat{v}_{i,t}(x_t, \hat{\theta}_{i,t}) - e^{x_t'\theta_i})}{1 + \sum_{i\in S_t} e^{x_t'\theta_i}} \\
\leq& \frac{2\alpha \sum_{i\in S_t} p_i ||x_t||_{\bar{V}_{i,t}^{-1}}}{1 + \sum_{i\in S_t} e^{x_t'\theta_i}} \\
\leq& \frac{2\alpha \max_{i\in\mathcal{N}} p_i}{\underline{\kappa}|S_t|} \sum_{i\in S_t} ||x_t||_{\bar{V}_{i,t}^{-1}},
\end{aligned}$$

where the first inequality is from Lemma A.3 in Agrawal et al. (2017a) and the optimality of S_t with respect to objective function $\hat{r}_t(S, x_t, \hat{\theta}_t)$, and the second and third inequalities are from event $\mathcal{E}_0$ such that

$$\hat{v}_{i,t}(x_t, \hat{\theta}_{i,t}) \geq \exp(x_t'\theta_i), \quad \text{and} \quad \exp(x_t'\hat{\theta}_{i,t}) - \exp(x_t'\theta_i) \leq \alpha||x_t||_{\bar{V}_{i,t}^{-1}}.$$

In Lemma 3.7.3 in the section of proofs, we will show that

$$\sum_{t\in\mathcal{T}} \sum_{i\in S_t} ||x_t||_{V_{i,t}^{-1}}/|S_t| = O(\sqrt{DNT\log T}).$$

This leads to

$$\sum_{t=1}^{T} (r(S_t^*, x_t, \theta) - r(S_t^*, x_t, \theta)) = O\left(\alpha\sqrt{DNT\log T}\right).$$

The proof of Theorem 3.3.1 is thus complete.

3.6.2 Outline of proof of Theorem 3.4.1

We split the regret into two parts: the projection regret, and the projected space regret. To that end, we define

$$\tilde{\theta}_i := M\theta_i, \quad \tilde{\theta}' = (\tilde{\theta}_1, \ldots, \tilde{\theta}_N).$$

Then

$$\begin{aligned} R(T) =& \sum_{t=1}^{T}(r(S_t^*, x_t, \theta) - r(S_t, x_t, \theta)) \\ =& \sum_{t=1}^{T}(r(S_t^*, x_t, \theta) - r(S_t^*, \tilde{x}_t, \tilde{\theta})) + \sum_{t=1}^{T}(r(S_t, \tilde{x}_t, \tilde{\theta}) - r(S_t, x_t, \theta)) \\ &+ \sum_{t=1}^{T}(r(S_t^*, \tilde{x}_t, \tilde{\theta}) - r(S_t, \tilde{x}_t, \tilde{\theta})). \end{aligned} \tag{3.13}$$

The first two terms on the right hand side is the projection regret, and the third term is the projected space regret.

Projection regret. Define the event

$$\mathcal{E}_1 := \big\{||\tilde{\theta}_i||_2 \leq \sqrt{1+\epsilon}R, ||\tilde{x}_t||_2 \leq \sqrt{1+\epsilon},\ \forall i \in \mathcal{N}, t \in \mathcal{T}\big\}. \tag{3.14}$$

In Lemma 3.7.8 of the section of proofs, we show that this event holds with probability at least $1-\delta$ when $d \geq 8\log((N+T)/\delta)/\epsilon^2$. This $\mathcal{E}_1$ essentially means that $\tilde{\theta}_i$ and $\tilde{x}_t$ are still in a bounded region after random projection. As a result, on $\mathcal{E}_1$

$$\begin{aligned} & r(S_t, \tilde{x}_t, \tilde{\theta}) - r(S_t, x_t, \theta) \\ =& \frac{\sum_{i\in S_t} p_i e^{\tilde{x}_t'\tilde{\theta}_i}(1+\sum_{i\in S_t} e^{x_t'\theta_i}) - \sum_{i\in S_t} p_i e^{x_t'\theta_i}(1+\sum_{i\in S_t} e^{\tilde{x}_t'\tilde{\theta}_i})}{(1+\sum_{i\in S_t} e^{\tilde{x}_t'\tilde{\theta}_i})(1+\sum_{i\in S_t} e^{x_t'\theta_i})} \\ =& \frac{\sum_{i\in S_t} p_i e^{\tilde{x}_t'\tilde{\theta}_i}\sum_{i\in S_t}(e^{x_t'\theta_i} - e^{\tilde{x}_t'\tilde{\theta}_i}) - (1+\sum_{i\in S_t} e^{\tilde{x}_t'\tilde{\theta}_i})\sum_{i\in S_t} p_i(e^{x_t'\theta_i} - e^{\tilde{x}_t'\tilde{\theta}_i})}{(1+\sum_{i\in S_t} e^{\tilde{x}_t'\tilde{\theta}_i})(1+\sum_{i\in S_t} e^{x_t'\theta_i})} \\ \leq& 2\max_{i\in\mathcal{N}} p_i \frac{\sum_{i\in S_t}|e^{x_t'\theta_i} - e^{\tilde{x}_t'\tilde{\theta}_i}|}{1+\sum_{i\in S_t} e^{x_t'\theta_i}} \leq \frac{2\max_{i\in\mathcal{N}} p_i}{\underline{\kappa}|S_t|}\sum_{i\in S_t}|e^{x_t'\theta_i} - e^{\tilde{x}_t'\tilde{\theta}_i}| \\ \leq& \frac{2\max_{i\in\mathcal{N}} p_i\overline{\kappa}^2}{\underline{\kappa}|S_t|}\sum_{i\in S_t}|x_t'\theta_i - \tilde{x}_t'\tilde{\theta}_i| \end{aligned}$$

where the last inequality is by Taylor's theorem and event $\mathcal{E}_1$ such that $\exp(\tilde{x}_t'\tilde{\theta}_i) \leq \exp(2R) =$

$\overline{\kappa}^2$. Summing over t yields

$$\sum_{t=1}^{T}(r(S_t,\tilde{x}_t,\tilde{\theta})-r(S_t,x_t,\theta)) \leq \frac{2\max_{i\in\mathcal{N}} p_i\overline{\kappa}^2}{\underline{\kappa}}\sum_{t=1}^{T}\sum_{i\in S_t}\frac{1}{|S_t|}|x_t'\theta_i-\tilde{x}_t'\tilde{\theta}_i|.$$

Now we define the second event

$$\mathcal{E}_2 := \big\{|x_t'\theta_i-\tilde{x}_t'\tilde{\theta}_i| \leq \epsilon/\gamma |x'_{t;>d_0}\theta_{i;>d_0}|_2, \quad \forall i\in\mathcal{N}, t\in\mathcal{T}\big\}. \tag{3.15}$$

In Lemma 3.7.9 of the section of proofs, we will show that event $\mathcal{E}_2$ holds with probability at least $1-\delta$ when $d \geq 8\log(TN/\delta)/\epsilon^2$. Then on this event, we have

$$\begin{aligned}\sum_{t=1}^{T}(r(S_t,\tilde{x}_t,\tilde{\theta})-r(S_t,x_t,\theta)) \leq& \frac{2\max_{i\in\mathcal{N}} p_i\overline{\kappa}^2}{\underline{\kappa}}\frac{\epsilon}{\gamma}\sum_{t=1}^{T}\sum_{i\in S_t}\frac{1}{|S_t|}|x'_{t;>d_0}\theta_{i;>d_0}| \\ \leq& \frac{2\max_{i\in\mathcal{N}} p_i\overline{\kappa}^2}{\underline{\kappa}}\frac{\epsilon}{\gamma}\sqrt{T\sum_{t=1}^{T}\sum_{i\in S_t}\frac{1}{|S_t|}|x'_{t;>d_0}\theta_{i;>d_0}|^2},\end{aligned}$$

where the first inequality is by the definition of even $\mathcal{E}_2$, and the last inequality is from Cauchy-Schwarz. Since $x_{t;>d_0}$ is the sparse portion of context x_t, its inner product with $\theta_{i;>d_0}$ can be bounded above, using Cauchy-Schwarz, by

$$|x'_{t;>d_0}\theta_{i;>d_0}| \leq ||x_{t;B_t}||_2||\theta_{i;B_t}||_2 \leq ||\theta_{i;B_t}||_2,$$

where $B_t := \{k > d_0 \ : x_{t;k} \neq 0\}$ is the support of the sparse portion of x_t. Thus, we have

$$\begin{aligned}\sum_{t=1}^{T}(r(S_t,\tilde{x}_t,\tilde{\theta})-r(S_t,x_t,\theta)) \leq& \frac{2\max_{i\in\mathcal{N}} p_i\overline{\kappa}^2}{\underline{\kappa}}\frac{\epsilon}{\gamma}\sqrt{T\sum_{t=1}^{T}\sum_{i\in S_t}\frac{1}{|S_t|}||\theta_{i;B_t}||_2^2} \\ \leq& \frac{2\max_{i\in\mathcal{N}} p_i\overline{\kappa}^2}{\underline{\kappa}}\frac{\epsilon}{\gamma}\sqrt{T\sum_{i\in\mathcal{N}}\sum_{t\in\mathcal{T}_i}||\theta_{i;B_t}||_2^2} \\ \leq& \frac{2R\max_{i\in\mathcal{N}} p_i\overline{\kappa}^2}{\underline{\kappa}}\frac{\epsilon}{\gamma}\sqrt{LNT},\end{aligned}$$

where the last inequality follows from the sparsity assumption on x_t.

Similar upper bound can be derived for $\sum_{t=1}^{T}(r(S_t^*,x_t,\theta)-r(S_t^*,\tilde{x}_t,\tilde{\theta}))$ and we omit

the details. As a result, the projection error can be bounded as

$$\sum_{t=1}^{T}(r(S_t^*, x_t, \theta) - r(S_t^*, \tilde{x}_t, \tilde{\theta})) + \sum_{t=1}^{T}(r(S_t, \tilde{x}_t, \tilde{\theta}) - r(S_t, x_t, \theta)) = O\left(\epsilon/\gamma\sqrt{LNT}\right). \tag{3.16}$$

Projected space regret. We next bound $\sum_{t=1}^{T}(r(S_t^*, \tilde{x}_t, \tilde{\theta}) - r(S_t, \tilde{x}_t, \tilde{\theta}))$. To this end, we first define the event

$$\mathcal{E}_3 := \left\{||\tilde{\theta}_i - \hat{\theta}_{i,t}||_{\bar{V}_{i,t}} \leq k_1' K^{3/2}(\epsilon/\gamma\sqrt{LN} + \sqrt{N(d_0+d)\log(NT/\delta)}),\ \forall i \in \mathcal{N}, t \in \mathcal{T}\right\}$$

for some constant k_1'. Then it follows from Lemma 3.7.10 in the section of proofs that event $\mathcal{E}_3$, on the events $\mathcal{E}_1$ and $\mathcal{E}_2$, holds with probability at least $1-\delta$. This event leads to an upper confidence bound which will be essential for our regret analysis.

The following derivations are all on the event $\mathcal{E}_1, \mathcal{E}_3$. On these events, we have that for all $i \in \mathcal{N}$ and $t \in \mathcal{T}$,

$$|e^{\tilde{x}_t'\hat{\theta}_{i,t}} - e^{\tilde{x}_t'\tilde{\theta}_i}| \leq \overline{\kappa}^2|\tilde{x}_t'\hat{\theta}_{i,t} - \tilde{x}_t'\tilde{\theta}_i| \leq \overline{\kappa}^2||\tilde{\theta}_i - \hat{\theta}_{i,t}||_{\bar{V}_{i,t}}||\tilde{x}_t||_{\bar{V}_{i,t}^{-1}} \leq \alpha||\tilde{x}_t||_{\bar{V}_{i,t}^{-1}},$$

where the second inequality follows from Cauchy-Schwarz, and the last inequality is from event $\mathcal{E}_3$ and appropriate choice of α (in particular, c). As a result, $\hat{v}_{i,t}(\tilde{x}_t, \hat{\theta}_{i,t}) \geq \exp(\tilde{x}_t'\tilde{\theta}_i)$ for all $i \in \mathcal{N}, t \in \mathcal{T}$, and

$$\begin{aligned}
r(S_t^*, \tilde{x}_t, \tilde{\theta}) - r(S_t, \tilde{x}_t, \tilde{\theta}) =& r(S_t^*, \tilde{x}_t, \tilde{\theta}) - \hat{r}_t(S_t, \tilde{x}_t, \hat{\theta}_t) + \hat{r}_t(S_t, \tilde{x}_t, \hat{\theta}_t) - r(S_t, \tilde{x}_t, \tilde{\theta}) \\
\leq& \hat{r}_t(S_t, \tilde{x}_t, \hat{\theta}_t) - r(S_t, \tilde{x}_t, \tilde{\theta}) \\
=& \frac{\sum_{i\in S_t} p_i \hat{v}_{i,t}(\tilde{x}_t, \hat{\theta}_{i,t})}{1 + \sum_{i\in S_t} \hat{v}_{i,t}(\tilde{x}_t, \hat{\theta}_{i,t})} - \frac{\sum_{i\in S_t} p_i e^{\tilde{x}_t'\tilde{\theta}_i}}{1 + \sum_{i\in S_t} e^{\tilde{x}_t'\tilde{\theta}_i}} \\
\leq& \frac{\sum_{i\in S_t} p_i(\hat{v}_{i,t}(\tilde{x}_t, \hat{\theta}_{i,t}) - e^{\tilde{x}_t'\tilde{\theta}_i})}{1 + \sum_{i\in S_t} e^{\tilde{x}_t'\tilde{\theta}_i}} \\
\leq& \frac{2\alpha \sum_{i\in S_t} p_i ||\tilde{x}_t||_{\bar{V}_{i,t}^{-1}}}{1 + \sum_{i\in S_t} e^{\tilde{x}_t'\tilde{\theta}_i}} \\
\leq& \frac{2\alpha \max_{i\in\mathcal{N}} p_i}{\underline{\kappa}|S_t|} \sum_{i\in S_t} ||\tilde{x}_t||_{\bar{V}_{i,t}^{-1}},
\end{aligned}$$

where the first inequality is from Lemma A.3 in Agrawal et al. (2017a) and the optimality of S_t with respect to objective function $\hat{r}_t(S, \tilde{x}_t, \hat{\theta}_t)$, and the second and third inequalities

follow from event $\mathcal{E}_3$ such that

$$\hat{v}_{i,t}(\tilde{x}_t, \hat{\theta}_{i,t}) \geq \exp(\tilde{x}_t'\tilde{\theta}_i), \qquad \exp(\tilde{x}_t'\hat{\theta}_{i,t}) - \exp(\tilde{x}_t'\tilde{\theta}_i) \leq \alpha||\tilde{x}_t||_{\bar{V}_{i,t}^{-1}}.$$

By Lemma 3.7.3 from the section of proofs modified to projected space, we have

$$\sum_{t\in\mathcal{T}}\sum_{i\in S_t} ||\tilde{x}_t||_{V_{i,t}^{-1}}/|S_t| = O(\sqrt{(d_0+d)NT\log T}).$$

Consequently,

$$\sum_{t=1}^{T}(r(S_t^*, x_t, \theta) - r(S_t^*, \tilde{x}_t, \tilde{\theta})) = O\left(\alpha\sqrt{(d_0+d)NT\log T}\right). \tag{3.17}$$

Then, combining (3.13), (3.16), and (3.17) gives us the desired result.

Note that the analysis above is based on three events $\mathcal{E}_1$, $\mathcal{E}_2$, $\mathcal{E}_3$. In the section of proofs we will prove that each of $\mathcal{E}_1$ and $\mathcal{E}_2$ holds with probability at least $1-\delta$, and $\mathcal{E}_3$ holds with probability at least $1-\delta$ given $\mathcal{E}_1 \cap \mathcal{E}_2$. Theorem 3.4.1 is then proved by union bound of (complements of) these events and applying law of total expectation (with $\delta = 1/T$ for example).

3.7 Proofs of Technical Results

This section provides all the proofs. Subsection 3.7.1 includes the technical results used in the proof of Theorem 3.3.1. Subsection 3.7.2 presents the proof of regret lower bound in Theorem 3.3.2. Subsection 3.7.3 derives the confidence bound used in proving Theorem 3.3.3. And finally, Subsection 3.7.4 contains some results for the proof of Theorem 3.4.1.

3.7.1 Technical lemmas for Theorem 3.3.1

Lemma 3.7.1 *Let $\delta > 0$ be an arbitrary positive real number, for all $t \in \mathcal{T}$,*

$$||\hat{\theta}_t - \theta||_{\bar{V}_t} \leq k_1 K\sqrt{ND\log(NT/\delta)}$$

for some constant k_1 with probability at least $1-\delta$.

Proof: Define the negative log-likelihood function at time period s as

$$l_s(\phi) := \log\left(1 + \sum_{i \in S_s} e^{x_s'\theta_i}\right) - \sum_{i \in S_s} Y_{i,s} x_s' \theta_i. \tag{3.18}$$

Then a maximum likelihood estimator $\hat{\theta}_t$ defined in (3.3) is also a minimum of the negative log-likelihood function $L_t(\phi) := -\log \mathcal{L}(\phi|\mathcal{F}_{t-1}) = \sum_{s=1}^{t-1} l_s(\phi)$, i.e., $\hat{\theta}_t \in \arg\min_{\phi \in \Theta} L_t(\phi)$.

Consider at time s, the gradient of negative log-likelihood function $\nabla l_s(\phi)$ is equal to

$$\nabla l_s(\phi) = \sum_{i \in S_s} (q_{i,s}(\phi) - Y_{i,s}) x_{i,s}, \tag{3.19}$$

where $q_{i,s}(\phi) = \exp(x_{i,s}'\phi)/(1 + \sum_{j \in S_s} \exp(x_{j,s}'\phi))$. We can compute its Hessian as

$$\begin{aligned}
\nabla^2 l_s(\phi) =& \frac{(1 + \sum_{j \in S_s} e^{x_{j,s}'\phi}) \sum_{i \in S_s} e^{x_{i,s}'\phi} x_{i,s} x_{i,s}' - (\sum_{i \in S_s} e^{x_{i,s}'\phi} x_{i,s})(\sum_{i \in S_s} e^{x_{i,s}'\phi} x_{i,s})'}{(1 + \sum_{j \in S_s} e^{x_{j,s}'\phi})^2} \\
\succeq & \frac{\sum_{i \in S_s} e^{x_{i,s}'\phi} x_{i,s} x_{i,s}'}{(1 + \sum_{j \in S_s} e^{x_{j,s}'\phi})^2} \\
= & \sum_i \frac{q_{i,s}(\phi)}{(1 + \sum_{j \in S_s} e^{x_{j,s}'\phi})} x_{i,s} x_{i,s}',
\end{aligned} \tag{3.20}$$

where the inequality is from Jensen's inequality. Here the notation $A \succeq B$ means that $A - B$ is a positive semidefinite matrix. Since $\hat{\theta}_t$ is an optimizer of the likelihood function, we must have

$$\begin{aligned}
0 \geq & L_t(\hat{\theta}_t) - L_t(\theta) \\
= & \nabla L_t(\theta)'(\hat{\theta}_t - \theta) + (\hat{\theta}_t - \theta)' \nabla^2 L_t(\bar{\theta}_t)(\hat{\theta}_t - \theta) \\
\geq & \sum_{s=1}^{t-1} \sum_{i \in S_s} (q_{i,s}(\theta) - Y_{i,s}) x_{i,s}'(\hat{\theta}_t - \theta) + \sum_{s=1}^{t-1} \sum_{i \in S_s} \frac{q_{i,s}(\bar{\theta}_t)}{(1 + \sum_{j \in S_s} e^{x_{j,s}'\bar{\theta}_t})} (x_{i,s}'(\hat{\theta}_t - \theta))^2,
\end{aligned}$$

where the equality is by Taylor's theorem with $\bar{\theta}_t$ being some point on the line segment connecting $\hat{\theta}_t$ and θ, and the second inequality is from (3.19) and (3.20). Now we analyze the two terms in the last summation separately. First, we consider the second term. We

have

$$\begin{aligned}\sum_{s=1}^{t-1}\sum_{i\in S_s}\frac{q_{i,s}(\bar{\theta}_t)}{(1+\sum_{j\in S_s}e^{x'_{j,s}\bar{\theta}_t})}(x'_{i,s}(\hat{\theta}_t-\theta))^2 \geq & \frac{\underline{\kappa}}{4\overline{\kappa}^2K}\sum_{s=1}^{t-1}\sum_{i\in S_s}\frac{1}{|S_s|}(x'_{i,s}(\hat{\theta}_t-\theta))^2\\ = & \frac{\underline{\kappa}}{4\overline{\kappa}^2K}||\hat{\theta}_t-\theta||^2_{V_t}\\ = & \frac{\underline{\kappa}}{4\overline{\kappa}^2K}||\hat{\theta}_t-\theta||^2_{\bar{V}_t}-\frac{\underline{\kappa}}{4\overline{\kappa}^2K}||\hat{\theta}_t-\theta||_2^2,\end{aligned}$$

where the first inequality is from boundedness of parameters and x_s and definition of $\overline{\kappa}, \underline{\kappa}$ in (3.2). For the first term, we have

$$\begin{aligned}\sum_{s=1}^{t-1}\sum_{i\in S_s}(q_{i,s}(\theta)-Y_{i,s})x'_{i,s}(\hat{\theta}_t-\theta) = & -\sum_{s=1}^{t-1}\sum_{i\in S_s}\epsilon_{i,s}x'_{i,s}(\hat{\theta}_t-\theta)\\ \geq & -|Z_t|,\end{aligned}$$

where

$$Z_t := \sum_{s=1}^{t-1}\sum_{i\in S_s}\epsilon_{i,s}x'_{i,s}(\hat{\theta}_t-\theta). \tag{3.21}$$

By Lemma 3.7.2, we have

$$|Z_t| \leq k_2\left(DN\log(NT/\delta)+\sqrt{DN\log(NT/\delta)}||\hat{\theta}_t-\theta||_{\bar{V}_t}\right)$$

for all $t \in \mathcal{T}$ with probability at least $1-\delta$, where k_2 is some constant. Combining the results above, we have

$$\frac{\underline{\kappa}}{4\overline{\kappa}^2K}||\hat{\theta}_t-\theta||^2_{\bar{V}_t}-\frac{\underline{\kappa}}{4\overline{\kappa}^2K}||\hat{\theta}_t-\theta||_2^2 \leq |Z_t|,$$

which is equivalent to

$$||\hat{\theta}_t-\theta||^2_{\bar{V}_t} \leq \frac{4\overline{\kappa}^2K}{\underline{\kappa}}|Z_t|+||\hat{\theta}_t-\theta||_2^2.$$

Now plugging in the value of (upper bound of) $|Z_t|$, we obtain, after some algebra,

$$\begin{aligned}||\hat{\theta}_t - \theta||_{\bar{V}_t} \leq & \frac{4\overline{\kappa}^2 K}{\underline{\kappa}} k_2 \sqrt{DN \log(NT/\delta)} \\ & + \sqrt{\frac{4\overline{\kappa}^2 K}{\underline{\kappa}} k_2 DN \log(NT/\delta) + 2R} \\ \leq & k_1 K \sqrt{DN \log(NT/\delta)}\end{aligned}$$

for some constant k_1 for all $t \in \mathcal{T}$. This completes the proof of Lemma 3.7.1. □

Lemma 3.7.2 *With probability at least $1-\delta$, for all $t \in \mathcal{T}$,*

$$|Z_t| \leq k_2 \left(DN \log(NT/\delta) + \sqrt{DN \log(NT/\delta)} ||\hat{\theta}_t - \theta||_{\bar{V}_t} \right)$$

for some constant k_2.

Proof: Define

$$Z_t(\phi) := \sum_{s=1}^{t-1} \sum_{i \in S_s} \epsilon_{i,s} x'_{i,s} (\phi - \theta).$$

Then it suffices to prove that with probability at least $1-\delta$,

$$|Z_t(\phi)| \leq k_2 \left(DN \log(NT/\delta) + \sqrt{DN \log(NT/\delta)} ||\phi - \theta||_{V_t} \right)$$

hold uniformly for all $t \in \mathcal{T}$ and $\phi \in \Theta$.

Let

$$M_s(\phi) := \sum_{i \in S_s} \epsilon_{i,s} x'_{i,s} (\phi - \theta),$$

which form a martingale difference sequence for $s \leq t-1$ satisfying

$$||M_s(\phi)||_2 \leq 4R.$$

Then we have

$$\begin{aligned}\mathbb{E}[M_s^2(\phi)|\mathcal{F}_{s-1}] &= \sum_{j \in S_s} q_{j,s}(\theta)(x'_{j,s}(\phi - \theta))^2 - \left(\sum_{j \in S_s} q_{j,s}(\theta) x'_{j,s}(\phi - \theta) \right)^2 \\ &\leq \sum_{j \in S_s} q_{j,s}(\theta)(x'_{j,s}(\phi - \theta))^2 \\ &\leq \overline{\kappa}/\underline{\kappa} \sum_{j \in S_s} \frac{1}{|S_s|} (x'_{j,s}(\phi - \theta))^2,\end{aligned}$$

which implies that, noting the definition of V_t in (3.9),

$$\sum_{s=1}^{t-1} \mathbb{E}[M_s^2(\phi)|\mathcal{F}_{s-1}] \leq \overline{\kappa}/\underline{\kappa}||\phi-\theta||_{V_t}^2 =: W_t(\phi).$$

According to Theorem A in Fan et al. (2015), we must have that for any $\phi \in \Theta$ and $\delta' > 0$,

$$\mathbb{P}\left(|\sum_{s=1}^{t-1} M_s(\phi)| > k_3\left(4R\log(1/\delta') + \sqrt{W_t(\phi)\log(1/\delta')}\right)\right) < \delta'$$

for some constant k_3. Now we consider a finite covering $\mathcal{H}(\upsilon)$ of Θ such that for any $\phi \in \Theta$, there exists a $\bar{\phi} \in \mathcal{H}(\upsilon)$ with $||\phi - \bar{\phi}||_2 \leq \upsilon$ (υ to be specified). Following the standard covering number argument (see e.g., Van der Vaart 1998), we must have $\log|\mathcal{H}(\upsilon)| \leq k_4 ND\log(R/\upsilon)$ for some constant k_4. As a result, let $\delta' = \delta/(T(R/\upsilon)^{k_4 DN})$, by a simple union bound, we have that with probability at least $1-\delta$, for any $t \in \mathcal{T}, \bar{\phi} \in \mathcal{H}(\upsilon)$,

$$|\sum_{s=1}^{t-1} M_s(\bar{\phi})| \leq k_5\left(DN\log(TR/(\upsilon\delta)) + \sqrt{W_t(\bar{\phi})DN\log(TR/(\upsilon\delta))}\right) \tag{3.22}$$

for some constant k_5. On the other hand, for any two $\phi, \bar{\phi}$ such that $||\phi - \bar{\phi}||_2 \leq \upsilon$, some simple algebra shows that

$$|\sum_s M_s(\phi) - \sum_s M_s(\bar{\phi})| \leq 2t\upsilon. \tag{3.23}$$

Applying Taylor's theorem, we obtain

$$|W_t(\phi) - W_t(\bar{\phi})| \leq 2\overline{\kappa}||V_t(\phi-\theta)||_2||\phi-\bar{\phi}||_2 \leq 4\overline{\kappa}RNt\upsilon, \tag{3.24}$$

where ϕ is some point between $\phi, \bar{\phi}$.

Let $\upsilon = \min(1/(2T), 1/(4\overline{\kappa}RNT))$, with probability at least $1-\delta$, for arbitrary $\phi \in \Theta$, let $\bar{\phi} \in \mathcal{H}(\upsilon)$ be the one in covering such that $||\phi - \bar{\phi}||_2 \leq \upsilon$, we have that, noting

$Z_t(\phi) = \sum_s M_s(\phi)$,

$$\begin{aligned}
|Z_t(\phi)| \leq & |Z_t(\bar{\phi})| + |Z_t(\bar{\phi}) - Z_t(\phi)| \\
\leq & k_5' \left(DN \log(NT/\delta) + \sqrt{W_t(\bar{\phi}) DN \log(NT/\delta)} \right) + 1 \\
\leq & k_5' \left(DN \log(NT/\delta) + \sqrt{(W_t(\phi) + |W_t(\phi) - W_t(\bar{\phi})|) DN \log(NT/\delta)} \right) + 1 \\
\leq & k_2 \left(DN \log(NT/\delta) + \sqrt{DN \log(NT/\delta)} ||\phi - \theta||_{V_t} \right),
\end{aligned}$$

where k_5' is some constant, the second inequality is from (3.22) and (3.23) with definition of ν, and the last inequality is from (3.24) with definition of ν. □

Lemma 3.7.3 *For an arbitrary sequence of x_t such that $||x_t||_2 \leq 1$ and S_t, we have*

$$\sum_{t \in \mathcal{T}} \sum_{i \in S_t} ||x_t||_{V_{i,t}^{-1}} / |S_t| = O(\sqrt{DNT \log T}).$$

Proof: By Cauchy-Schwarz, we have

$$\sum_{t \in \mathcal{T}} \sum_{i \in S_t} ||x_t||_{V_{i,t}^{-1}} / |S_t| \leq \sqrt{T \sum_{t \in \mathcal{T}} \sum_{i \in S_t} ||x_t||^2_{V_{i,t}^{-1}} / |S_t|}. \tag{3.25}$$

Note that

$$\begin{aligned}
\det(\bar{V}_{t+1}) = & \det \left(\bar{V}_t + \sum_{i \in S_t} x_{i,t} x_{i,t}' / |S_t| \right) \\
= & \det(\bar{V}_t) \det \left(I + \sum_{i \in S_t} \bar{V}_t^{-1/2} x_{i,t} (\bar{V}_t^{-1/2} x_{i,t})' / |S_t| \right).
\end{aligned}$$

Since

$$\sum_{i \in S_t} \bar{V}_t^{-1/2} x_{i,t} (\bar{V}_t^{-1/2} x_{i,t})' / |S_t|$$

is a positive semidefinite matrix with rank at most $|S_t|$, we assume without loss of generality that its eigenvalues are $\lambda_1 \geq \ldots \geq \lambda_{|S_t|}$, which are all nonnegative (note that if $|S_t| > D$,

we let $\lambda_k = 0$ for any $k > D$). Then,

$$\begin{aligned}
&\det\left(I + \sum_{i\in S_t} \bar{V}_t^{-1/2} x_{i,t} (\bar{V}_t^{-1/2} x_{i,t})'/|S_t|\right) \\
=&\prod_{k=1}^{|S_t|}(1+\lambda_k) \\
\geq& 1 + \sum_{k=1}^{|S_t|} \lambda_k \\
=& 1 + \operatorname{tr}\left(\sum_{i\in S_t} \bar{V}_t^{-1/2} x_{i,t} (\bar{V}_t^{-1/2} x_{i,t})'/|S_t|\right) \\
=& 1 + \sum_{i\in S_t} ||x_{i,t}/\sqrt{S_t}||^2_{\bar{V}_t^{-1}} \\
=& 1 + ||X_t||^2_{\bar{V}_t^{-1}},
\end{aligned}$$

where

$$X_t := \sum_{i\in S_t} x_{i,t}/\sqrt{|S_t|}.$$

As a result, by an iterative argument, we obtain that

$$\det(\bar{V}_{T+1}) \geq \det(I) \prod_{s=1}^{T} \left(1 + ||X_s||^2_{\bar{V}_s^{-1}}\right),$$

which is equivalent to

$$\log\det(\bar{V}_{T+1}) \geq \log\det(I) + \sum_{s=1}^{T} \log\left(1 + ||X_s||^2_{\bar{V}_s^{-1}}\right).$$

Since $||X_t||_2 \leq \sqrt{\sum_{i\in S_s} ||x_t||_2^2/|S_t|} \leq 1$, we must have $||X_s||^2_{\bar{V}_s^{-1}} \leq 1$. Because $x \leq 2\log(1+x)$ for all $x \in [0,1]$, it follows that $2\log\left(1 + ||X_s||^2_{\bar{V}_s^{-1}}\right) \geq ||X_s||^2_{\bar{V}_s^{-1}}$. Consequently,

$$\sum_{s=1}^{T} ||X_s||^2_{\bar{V}_s^{-1}} \leq 2\log\det(\bar{V}_{T+1}) - 2\log\det(I) \leq 2DN\log(1+T/D),$$

where the last inequality is from Lemma 10 in Abbasi-Yadkori et al. (2011), and this, by plugging in (3.25), completes the proof. □

3.7.2 Proof of Theorem 3.3.2 on lower bound

The proof is to connect our problem to an ordinary MNL bandit instance. To do that, we first split the time horizon T into $D-1$ groups. In each group, we have $T' = \lceil T/(D-1) \rceil$ time periods (except the last group). Index these $D-1$ groups by $g = 1, \ldots, D-1$.

The first step to create the bandit problem instance is to construct the sequence of x_t. Let $x_{t;k} = 1$ for all $k = \lceil t/T' \rceil$ and $t \in \mathcal{T}$; otherwise, $x_{t;k} = 0$. This construction is legitimate in two aspects. First, for any t, k such that $x_{t;k} = 1$, we must have $k \leq D$ since $k = \lceil t/T' \rceil \leq \lceil T/T' \rceil \leq D-1$ so that the index is within the range of D. Second, we obviously have $||x_t||_2 = 1$ for all $t \in \mathcal{T}$.

The second step is to create θ_i for all $i \in \mathcal{N}$. We set $\theta_{i;k} = \log(1+\epsilon)$ for all $i \in \mathcal{N}$ and $k = 1, \ldots, D$ such that $\lceil i/K \rceil = k$. Otherwise $\theta_{i;k} = 0$. To better understand this construction, we have for $i = 1, \ldots, K$, $\theta_{i,1} = \log(1+\epsilon)$; for $i = K+1, \ldots, 2K$, $\theta_{i,2} = \log(1+\epsilon)$;...; for $i = (D-1)K, \ldots, DK$, $\theta_{i,D} = \log(1+\epsilon)$. Again, this construction is legitimate because $N \geq DK$, and by taking $\epsilon = \sqrt{N/(144T')}$, we have $||\theta_i||_2 \leq \epsilon \leq R$ because $T \geq ND/(144R^2)$.

We say that a time period t belongs to group g if $\lceil t/T' \rceil = g$. By the construction of x_t and θ above, we have that for all time periods in each group g, the optimal assortment is exactly $i = (g-1)K+1, \ldots, gK$. So for each group g, we have an independent MNL bandit instance in that the utility for products in optimal assortment are exactly $e^{x_t'\theta_i} = 1+\epsilon$, and otherwise it is 1. Then by taking value of $\epsilon = \sqrt{N/(144T')}$, and since $N \geq 4K$ (by assumption) and $T' \geq N/144$ (since $T \geq ND/144$), we apply results in Corollary 3.7.1, to be proved below, to obtain that the expected regret of each group is bounded below by $\sqrt{NT'}/(324K)$. As a result, the expected regret of our problem is at least

$$(D-2)\frac{\sqrt{NT'}}{324K} = \Omega\left(\frac{\sqrt{DNT}}{K}\right),$$

and we obtain the desired result.

Adopting the techniques in Chen and Wang (2017), we derive a lower bound for MNL bandit problem when each v_i are bounded and independent of K. The construction used in Chen and Wang (2017) assumes $v_i = \Theta(1/K)$, and it does not fit our case since we assume all parameters are bounded and independent of K. To resolve this issue, we consider the following problem instance. Let S^* be an assortment with $|S^*| = K$, we construct the bandit instance as $v_i = 1+\epsilon$ for all $i \in S^*$ where ϵ is to be specified; otherwise $v_i = 1$. For all products, $p_i = 1$. Then we have the following result similar to Theorem 1 of Chen and Wang (2017), from which Theorem 3.3.2 follows.

Proposition 3.7.1 *For bandit problems with parameters N, K, T, suppose $N/4 \geq K$ and $T \geq N/144$, then for any policy π,*

$$\mathbb{E}[R^{\pi}(T)] \geq \frac{\sqrt{NT}/K}{324},$$

where the expectation is taken with respect to both the randomness of the problem and the uniformly generated optimal assortment S^*.

Proof: The proof is very similar to that of Theorem 1 in Chen and Wang (2017), with the only difference being $v_i \in \{1, 1+\epsilon\}$ while $v_i \in \{1/K, (1+\epsilon)/K\}$ in Chen and Wang (2017). Therefore, we will only discuss the parts that are different.

Using notation in Chen and Wang (2017), we let $\mathbb{E}_{S^*}[\cdot]$ denote the expectation given optimal assortment S^* with $S^* \in \mathcal{S}_K := \{S \in \mathcal{S} : |S| = K\}$, and $r_v(S)$ denote the expected revenue of assortment S given utility parameters v. Following the proof of Lemma 1 in Chen and Wang (2017), we can get that in our problem instance, for arbitrary S_t with $|S_t| = K$,

$$r_v(S^*) - r_v(S_t) \geq \frac{\delta\epsilon}{9K},$$

where $\delta = 1 - |S^* \cap S_t|/K$. Then following Chen and Wang (2017) and noting that $T_i := T_i(T)$ in this chapter is equivalent to N_i in Chen and Wang (2017), we obtain

$$\begin{aligned}\mathbb{E}[R^{\pi}(T)] =& \frac{1}{|\mathcal{S}_K|}\sum_{S^*\in\mathcal{S}_K}\mathbb{E}_{S^*}\left[r_v(S^*) - r_v(S_t)\right] \\ \geq& \frac{\epsilon}{9K}\left(T - \frac{1}{|\mathcal{S}_K|}\sum_{S^*\in\mathcal{S}_K}\frac{1}{K}\sum_{i\in S^*}\mathbb{E}_{S^*}[T_i]\right).\end{aligned} \tag{3.26}$$

The next step is to find an upper bound of $\frac{1}{|\mathcal{S}_K|}\sum_{S^*\in\mathcal{S}_K}\frac{1}{K}\sum_{i\in S^*}\mathbb{E}_{S^*}[T_i]$. Following the same argument, we obtain that

$$\begin{aligned}\frac{1}{|\mathcal{S}_K|}\sum_{S^*\in\mathcal{S}_K}\frac{1}{K}\sum_{i\in S^*}\mathbb{E}_{S^*}[T_i] \leq& \frac{1}{K}\sum_{i\in\mathcal{N}}\frac{1}{|\mathcal{S}_K|}\sum_{S'\in\mathcal{S}_{K-1}^{(i)}}\left(\mathbb{E}_{S'}[T_i] + T\sqrt{\frac{1}{2}KL(P||Q)}\right) \\ \leq& \frac{T}{3} + \frac{T}{K}\sum_{i\in\mathcal{N}}\frac{1}{|\mathcal{S}_K|}\sum_{S'\in\mathcal{S}_{K-1}^{(i)}}\sqrt{\frac{1}{2}KL(P||Q)},\end{aligned} \tag{3.27}$$

where

$$\mathcal{S}_{K-1}^{(i)} := \{S \in \mathcal{S}_{K-1} : i \notin S\}, \quad P = P_{S'}, Q = P_{S'\cup\{i\}}$$

are the distribution generated by the algorithm given optimal assortment S', $S' \cup \{i\}$, and $KL(\cdot||\cdot)$ is the Kullback-Leibler (KL) divergence. Now we need to bound the KL divergence, and the upper bound is proved in Lemma 2 in Chen and Wang (2017). The results will be slightly different, so we present the process here. First, for any S_t with $|S_t| \leq K$ and $i \notin S_t$, the KL divergence of each period t conditioned on S_t is zero. So we only focus on t such that $i \in S_t$. Define $K' = |S_t|$ and $J = |S_t \cap S'|$, we must have that the probabilities a customer chooses product $j \in S_t \cup \{0\}$ at time t under P, Q are

$$\begin{aligned} p_j =& \frac{v_j}{1 + K' + J\epsilon}, \\ q_j =& \frac{v_j}{1 + K' + (J+1)\epsilon}. \end{aligned}$$

Therefore,

$$|p_0 - q_0| = \left| \frac{1}{1 + K' + J\epsilon} - \frac{1}{1 + K' + (J+1)\epsilon} \right| \leq \frac{\epsilon}{(1+K')^2}.$$

For $j \in S_t$ such that $j \neq i$,

$$|p_j - q_j| \leq (1+\epsilon) \left| \frac{1}{1 + K' + J\epsilon} - \frac{1}{1 + K' + (J+1)\epsilon} \right| \leq \frac{2\epsilon}{(1+K')^2}.$$

Then we obtain

$$|p_i - q_i| = \left| \frac{1}{1 + K' + J\epsilon} - \frac{1+\epsilon}{1 + K' + (J+1)\epsilon} \right| \leq \frac{2\epsilon}{(1+K')}.$$

Note that $q_j \geq 1/(2(1+K'))$ for all j. Following Lemma 3 in Chen and Wang (2017), we have

$$KL(P(\cdot|S_t)||Q(\cdot||S_t)) \leq \frac{2\epsilon^2}{(1+K')^3} + \frac{8(K'-1)\epsilon^2}{(1+K')^3} + \frac{8\epsilon^2}{(1+K')} \leq \frac{16\epsilon^2}{1+|S_t|},$$

which gives us

$$KL(P||Q) \leq \mathbb{E}_{S'} \left[\sum_{t \in \mathcal{T}_i(T)} \frac{16\epsilon^2}{1+|S_t|} \right].$$

Then following the analysis in Section 3.4 of Chen and Wang (2017), we have

$$\begin{aligned}
\frac{T}{K}\sum_{i\in\mathcal{N}}\frac{1}{|\mathcal{S}_K|}\sum_{S'\in\mathcal{S}_{K-1}^{(i)}}\sqrt{\frac{1}{2}KL(P||Q)} \leq & T\max_{S'\in\mathcal{S}_{K-1}}\sqrt{\frac{1}{2(N-K+1)}\sum_{i\notin S'}KL(P||Q)} \\
\leq & T\max_{S'\in\mathcal{S}_{K-1}}\sqrt{\frac{1}{2(N-K+1)}\sum_{i\in\mathcal{N}}\mathbb{E}_{S'}\left[\sum_{t\in\mathcal{T}_i}\frac{16\epsilon^2}{1+|S_t|}\right]} \\
= & T\max_{S'\in\mathcal{S}_{K-1}}\sqrt{\frac{1}{2(N-K+1)}\mathbb{E}_{S'}\left[\sum_{t\in\mathcal{T}}\sum_{i\in S_t}\frac{18\epsilon^2}{1+|S_t|}\right]} \\
\leq & T\max_{S'\in\mathcal{S}_{K-1}}\sqrt{\frac{16\epsilon^2 T}{2(N-K+1)}} \leq T\sqrt{16\epsilon^2 T/N}.
\end{aligned}$$

Combining this inequality with (3.26) and (3.27), we have that

$$\mathbb{E}[R^\pi(T)] \geq \frac{\epsilon}{9K}\left(\frac{2T}{3} - T\sqrt{16\epsilon^2 T/N}\right).$$

Taking $\epsilon = \sqrt{N/(144T)} \leq 1$, we obtain the desired result. □

3.7.3 Online Newton Step for Theorem 3.3.3

The proof of Theorem 3.3.3 is a simple combination of Theorem 3.3.1 and the following result on the confidence bound of $\check{\theta}_t$.

Lemma 3.7.4 *The regret of online Newton step is bounded above by*

$$\bar{\beta}_t = k_6(d_0+d)NK^2\log t + k_7/K$$

for some constant k_6, k_7. That is, for any $t \in \mathcal{T}$,

$$\sum_{s=1}^{t}\left(l_s(\bar{\theta}_s, x_s) - l_s(\theta, x_s)\right) \leq \bar{\beta}_t.$$

Proof: Note that $l_s(\bar{\theta}_s, x_s)$ can be expressed as a function of $\bar{z}_{i,s} = x_s'\bar{\theta}_{i,s}$ for all $i \in S_s$ and Y_s representing customer's choice at time s, i.e.,

$$l_s(\bar{\theta}_s, x_s) = l_s(\bar{z}_s, Y_s).$$

Similarly, we can write

$$l_s(\theta, x_s) = l_s(z_s, Y_s),$$

where $z_s' = (z_{i_1,s}, \ldots, z_{i_{|S_s|},s})$ with $z_{i,s} = x_s'\theta_i$ and $i_1, \ldots, i_{|S_s|}$ represent the indices of products in S_s. One can verify that the Hessian of l_s with respect to z_s is

$$\begin{aligned}\nabla_z^2 l_s(z_s, Y_s) =& \frac{\text{diag}(e^{z_s})(1 + \sum_{i\in S_s} e^{z_{i,s}}) - e^{z_s}(e^{z_s})'}{(1 + \sum_{i\in S_s} e^{z_{i,s}})^2} \\ \succeq& \frac{\text{diag}(e^{z_s})}{(1 + \sum_{i\in S_s} e^{z_{i,s}})^2},\end{aligned}$$

where $(e^{z_s})' := (e^{z_{i_1,s}}, \ldots, e^{z_{i_{|S_s|},s}})$, and $\text{diag}(e^{z_s})$ is a diagonal matrix with diagonal entries e^{z_s}. The inequality above is from Jensen's inequality. Let $1/\Gamma_s = \underline{\kappa}/(1 + \overline{\kappa}|S_s|)^2$ be the lower bound of $\frac{e^{z_{j,s}}}{(1+\sum_{i\in S_s} e^{z_{i,s}})^2}$ for all $j \in S_s$. We see that $l_s(z_s, Y_s)$ is a $1/\Gamma_s$-strongly convex function of z_s, thus

$$\begin{aligned}l_s(\bar{z}_s, Y_s) - l_s(z_s, Y_s) \leq& \nabla l_s(\bar{z}_s, Y_s)'(\bar{z}_s - z_s) - \frac{1}{2\Gamma_s}||\bar{z}_s - z_s||_2^2 \\ =& \sum_{i\in S_s} \partial_i l_s(\bar{z}_s, Y_s)(x_s'\bar{\theta}_{i,s} - x_s'\theta_i) - \frac{|S_s|}{2\Gamma_s}\sum_{i\in S_s}(x_s'\bar{\theta}_{i,s} - x_s'\theta_i)^2/|S_s| \\ \leq& \sum_{i\in S_s} \partial_i l_s(\bar{z}_s, Y_s)(x_s'\bar{\theta}_{i,s} - x_s'\theta_i) - \frac{1}{2\Gamma_0}\sum_{i\in S_s}(x_s'\bar{\theta}_{i,s} - x_s'\theta_i)^2/|S_s|,\end{aligned} \tag{3.28}$$

where $\partial_i l_s(\bar{z}_s, Y_s)$ is the partial derivative with respect to $\bar{z}_{i,s}$, and $\Gamma_0 := (\overline{\kappa}^2 K + 2\overline{\kappa} + 1)/\underline{\kappa}$. Note that by the updating rule in algorithm OL-NEW,

$$\bar{\theta}_{s+1}^0 - \theta = \bar{\theta}_s - \theta - \Gamma_0 \bar{V}_{s+1}^{-1} \sum_{i\in S_s} c_{i,s} x_{i,s}$$

where $c_{i,s} := \partial_i l_s(\bar{z}_s, Y_s)$. This implies

$$||\bar{\theta}_{s+1}^0 - \theta||_{\bar{V}_{s+1}}^2 = ||\bar{\theta}_s - \theta||_{\bar{V}_{s+1}}^2 - 2\Gamma_0 \sum_{i\in S_s} c_{i,s} x_{i,s}'(\bar{\theta}_s - \theta) + \Gamma_0^2 ||\sum_{i\in S_s} c_{i,s} x_{i,s}||_{\bar{V}_{s+1}^{-1}}^2. \tag{3.29}$$

By the property of the generalized projection (Hazan et al. 2007), $||\bar{\theta}^0_{s+1}-\theta||^2_{\bar{V}_{s+1}} \geq ||\bar{\theta}_{s+1}-\theta||^2_{\bar{V}_{s+1}}$. Combining with (3.29), we get

$$\sum_{s=1}^{t}\sum_{i\in S_s} c_{i,s}x'_{i,s}(\bar{\theta}_s-\theta) \leq \sum_{s=1}^{t}\frac{\Gamma_0}{2}||\sum_{i\in S_s} c_{i,s}x_{i,s}||^2_{\bar{V}^{-1}_{s+1}} + \sum_{s=1}^{t}\frac{1}{2\Gamma_0}\left(||\bar{\theta}_s-\theta||^2_{\bar{V}_{s+1}} - ||\bar{\theta}_{s+1}-\theta||^2_{\bar{V}_{s+1}}\right).$$

Note that the second sum on the right-hand side can be bounded as

$$\begin{aligned}
&\sum_{s=1}^{t}\frac{1}{2\Gamma_0}\left(||\bar{\theta}_s-\theta||^2_{\bar{V}_{s+1}} - ||\bar{\theta}_{s+1}-\theta||^2_{\bar{V}_{s+1}}\right)\\
\leq&\frac{1}{2\Gamma_0}||\bar{\theta}_1-\theta||^2_{\bar{V}_2} + \sum_{s=2}^{t}\frac{1}{2\Gamma_0}\left(||\bar{\theta}_s-\theta||^2_{\bar{V}_{s+1}} - ||\bar{\theta}_s-\theta||^2_{\bar{V}_s}\right)\\
=&\frac{1}{2\Gamma_0}||\bar{\theta}_1-\theta||^2_{\bar{V}_2} - \frac{1}{2\Gamma_0}||\bar{\theta}_1-\theta||^2_{\sum_{i\in S_1}x_{i,1}x'_{i,1}/|S_1|} + \sum_{s=1}^{t}\frac{1}{2\Gamma_0}||\bar{\theta}_s-\theta||^2_{\sum_{i\in S_s}x_{i,s}x'_{i,s}/|S_s|}\\
=&\frac{1}{2\Gamma_0}||\bar{\theta}_1-\theta||^2_2 + \sum_{s=1}^{t}\frac{1}{2\Gamma_0}||\bar{\theta}_s-\theta||^2_{\sum_{i\in S_s}x_{i,s}x'_{i,s}/|S_s|}\\
\leq&2R^2/\Gamma_0 + \sum_{s=1}^{t}\frac{1}{2\Gamma_0}||\bar{\theta}_s-\theta||^2_{\sum_{i\in S_s}x_{i,s}x'_{i,s}/|S_s|},
\end{aligned}$$

where the first equality is from the definition of $\bar{V}_s$. Therefore, we have

$$\sum_{s=1}^{t}\sum_{i\in S_s} c_{i,s}x'_{i,s}(\bar{\theta}_s-\theta) \leq \sum_{s=1}^{t}\frac{\Gamma_0}{2}||\sum_{i\in S_s} c_{i,s}x_{i,s}||^2_{\bar{V}^{-1}_{s+1}} + k_7/K + \sum_{s=1}^{t}\frac{1}{2\Gamma_0}||\bar{\theta}_s-\theta||^2_{\sum_{i\in S_s}x_{i,s}x'_{i,s}/|S_s|}$$

for $k_7/K \geq 2R^2/\Gamma_0$. Rearranging this inequality, we have

$$\sum_{s=1}^{t}\sum_{i\in S_s} c_{i,s}x'_{i,s}(\bar{\theta}_s-\theta) - \sum_{s=1}^{t}\frac{1}{2\Gamma_0}||\bar{\theta}_s-\theta||^2_{\sum_{i\in S_s}x_{i,s}x'_{i,s}/|S_s|} \leq \sum_{s=1}^{t}\frac{\Gamma_0}{2}||\sum_{i\in S_s} c_{i,s}x_{i,s}||^2_{\bar{V}^{-1}_{s+1}} + k_7/K.$$

Combining with (3.28), we get

$$\begin{aligned}
\sum_{s=1}^{t}(l_s(\bar{z}_s,Y_s) - l_s(z_s,Y_s)) \leq& \sum_{s=1}^{t}\frac{\Gamma_0}{2}||\sum_{i\in S_s} c_{i,s}x_{i,s}||^2_{\bar{V}^{-1}_{s+1}} + k_7/K\\
=&\sum_{s=1}^{t}\frac{\Gamma_0}{2}\sum_{i\in S_s} c^2_{i,s}||x_s||^2_{\bar{V}^{-1}_{i,s+1}} + k_7/K,
\end{aligned}$$

where the inequality follows from the definition of $x_{i,s}$ in (3.8) and $\bar{V}_s$ in (3.9). To bound the right-hand side of the inequality above, we note that $c_{i,s}^2 = (q_{i,s}(\bar{\theta}_s) - Y_{i,s})^2 \leq 1$, hence

$$\sum_{s=1}^{t} \frac{\Gamma_0}{2} \sum_{i \in S_s} c_{i,s}^2 ||x_s||^2_{\bar{V}_{i,s+1}^{-1}} \leq \sum_{s=1}^{t} \frac{\Gamma_0 |S_s|}{2} \sum_{i \in S_s} ||x_s||^2_{\bar{V}_{i,s+1}^{-1}} / |S_s| \leq k_6 DNK^2 \log t$$

for some constant k_6, where the last inequality is from the proof of Lemma 3.7.3 and the fact that $\bar{V}_{i,s+1} \succeq \bar{V}_{i,s}$. As a result, we have

$$\sum_{s=1}^{t} (l_s(\bar{z}_s, Y_s) - l_s(z_s, Y_s)) \leq k_6 DNK^2 \log t + k_7/K = \bar{\beta}_t,$$

and the proof is complete. □

Lemma 3.7.4 presents the regret of the online Newton step with estimated parameters $\bar{\theta}_s$ for $s = 1, \ldots, t$. Note that these parameters are not directly used to construct the upper confidence bounds. As seen in Section 3.3.3, $\bar{\theta}_s$ are used to construct another parameter $\check{\theta}_t$ and we need to prove that $\check{\theta}_t$ has small confidence bound. The next lemma presents this confidence bound result which is obtained based on the regret in Lemma 3.7.4.

Lemma 3.7.5 *We have that with probability at least $1 - \delta$, it holds that*

$$||\theta - \check{\theta}_t||_{\bar{V}_t} \leq \bar{\alpha}_T = O(K^{3/2} \sqrt{DN \log(T/\delta)}).$$

Proof: By Taylor's theorem and Lemma 3.7.4, we have

$$\begin{aligned} \bar{\beta}_t \geq & \sum_{s=1}^{t} (l_s(\bar{z}_s, Y_s) - l_s(z_s, Y_s)) \geq \sum_{s=1}^{t} \nabla l_s(z_s, Y_s)'(\bar{z}_s - z_s) + \sum_{s=1}^{t} \frac{1}{2\Gamma_s} ||\bar{z}_s - z_s||_2^2 \\ = & \sum_{s=1}^{t} \sum_{i \in S_s} (q_{i,s}(\theta) - Y_{i,s}) x_s'(\bar{\theta}_{i,s} - \theta_i) + \sum_{s=1}^{t} \frac{|S_s|}{2\Gamma_s} \sum_{i \in S_s} (x_s'(\bar{\theta}_{i,s} - \theta_i))^2 / |S_s|. \end{aligned} \tag{3.30}$$

Since $q_{i,s}(\theta) - Y_{i,s} = -\epsilon_{i,s}$, we have

$$\sum_{i \in S_s} (q_{i,s}(\theta) - Y_{i,s}) x_s'(\bar{\theta}_{i,s} - \theta_i) = -\sum_{i \in S_s} \epsilon_{i,s} . x_{i,s}'(\bar{\theta}_s - \theta).$$

Let $\mu_{i,s} = x_{i,s}'(\bar{\theta}_s - \theta)$. Then $\sum_{s=1}^{t} \sum_{i \in S_s} \epsilon_{i,s} x_{i,s}'(\bar{\theta}_s - \theta) = \sum_{s=1}^{t} \sum_{i \in S_s} \epsilon_{i,s} \mu_{i,s}$. Note that for each s, we can write $\mu_s' := (\mu_{i_1,s}, \ldots, \mu_{i_{|S_s|},s})$ and $\epsilon_s' := (\epsilon_{i_1,s}, \ldots, \epsilon_{i_{|S_s|},s})$, where $i_1, \ldots, i_{|S_s|}$ are the indices of products in S_s. Then we have $\sum_{s=1}^{t} \sum_{i \in S_s} \epsilon_{i,s} \mu_{i,s} = \sum_{s=1}^{t} \epsilon_s' \mu_s$.

To develop a concentration inequality, we follow Theorem 7 and Corollary 8 in Abbasi-Yadkori et al. (2012) to define

$$\begin{aligned} D_t^\lambda &= \exp(\lambda \epsilon_t' \mu_t/2 - \lambda^2 \mu_t' \mu_t/2), \\ S_t &= \sum_{s=1}^{t} \epsilon_t' \mu_t, \\ M_t^\lambda &= \exp\left(\lambda S_t/2 - \lambda^2 \sum_{s=1}^{t} \mu_t' \mu_t/2\right). \end{aligned}$$

By the sub-Gaussianity of $\epsilon_t' \mu_t$ and $|\epsilon_t' \mu_t| \leq 2 \max_{i \in S_t} |\mu_{i,t}|$, we have, conditioning on history,

$$\mathbb{E}[D_t^\lambda] \leq \exp(\lambda^2 \max_{i \in S_s} |\mu_{i,t}|^2/2 - \lambda^2 \mu_t' \mu_t/2) \leq 1.$$

Then following the proof of Theorem 7 and Corollary 8 of Abbasi-Yadkori et al. (2012), we can show that for all t, with probability at least $1-\delta$,

$$\sum_{s=1}^{t} \sum_{i \in S_s} \epsilon_{i,s} x_{i,s}' (\bar{\theta}_s - \theta) \leq 2 \sqrt{2 \left(1 + \sum_{s=1}^{t} \sum_{i \in S_s} \mu_{i,s}^2\right) \log\left(\sqrt{1 + \sum_{s=1}^{t} \sum_{i \in S_s} \mu_{i,s}^2}/\delta\right)}. \quad (3.31)$$

Combining (3.30) and (3.31), we have that with probability at least $1-\delta$, for all $t \in \mathcal{T}$,

$$\begin{aligned} \sum_{s=1}^{t} \sum_{i \in S_s} \mu_{i,s}^2/|S_s| \leq & k_9 K \bar{\beta}_t \\ & + k_9 K^{3/2} \sqrt{2 \left(1 + \sum_{s=1}^{t} \sum_{i \in S_s} \mu_{i,s}^2/|S_s|\right) \log\left(\sqrt{1 + \sum_{s=1}^{t} \sum_{i \in S_s} \mu_{i,s}^2/|S_s|}/\delta\right)} \end{aligned}$$

for some constant k_9. This implies, using Lemma 2 in Jun et al. (2017), that

$$\sum_{s=1}^{t} \sum_{i \in S_s} \mu_{i,s}^2/|S_s| \leq k_{10} K \bar{\beta}_t + k_{10} K^3 \log(T/\delta) \quad (3.32)$$

for some constant k_{10}. Then one can rewrite the above as

$$\begin{aligned} & ||\bar{\boldsymbol{z}}_t - \boldsymbol{X}_t \theta||_2^2 \leq k_{10} K \bar{\beta}_t + k_{10} K^3 \log(T/\delta) \\ \iff & ||\bar{\boldsymbol{z}}_t - \boldsymbol{X}_t \theta||_2^2 + ||\theta||_2^2 \leq k_{10} K \bar{\beta}_t + k_{10} K^3 \log(T/\delta) + ||\theta||_2^2 \\ \implies & ||\check{\theta}_t^0 - \theta||_{\bar{V}_t}^2 + ||\bar{\boldsymbol{z}}_t||_2^2 - (\check{\theta}_t^0)' \boldsymbol{X}_t' \bar{\boldsymbol{z}}_t \leq k_{10} K \bar{\beta}_t + k_{10} K^3 \log(T/\delta) + R^2, \end{aligned}$$

where the last inequality is by

$$\check{\theta}_t^0 = \bar{V}_t^{-1} \boldsymbol{X}_t' \bar{\boldsymbol{z}}_t = \arg\min_{\phi} ||\bar{\boldsymbol{z}}_t - \boldsymbol{X}_t \phi||_2^2 + ||\phi||_2^2.$$

Note that from the definition of $\check{\theta}_t^0$ and $\boldsymbol{X}_t' \boldsymbol{X}_t = V_t \preceq \bar{V}_t$, we have

$$||\bar{\boldsymbol{z}}_t||_2^2 - (\check{\theta}_t^0)' \boldsymbol{X}_t' \bar{\boldsymbol{z}}_t \geq 0.$$

Therefore,

$$||\check{\theta}_t^0 - \theta||_{\bar{V}_t}^2 \leq k_{10} K \bar{\beta}_t + k_{10} K^3 \log(T/\delta) + R^2 =: \bar{\alpha}_t^2.$$

Finally, we note the above inequality also holds by replacing $\check{\theta}_t^0$ by $\check{\theta}_t$ according to the generalized projection property (Lemma 8 in Hazan et al. 2007). Thus the desired result is proved. □

3.7.4 Technical lemmas for Theorem 3.4.1

In this subsection, we present the technical results used in the proof of Theorem 3.4.1. The first two lemmas can be found in Kaban (2015).

Lemma 3.7.6 (Johnson-Lindenstrauss lemma) *Let x be a vector in $\mathbb{R}^D$ and $M \in \mathbb{R}^{d \times D}$ a random matrix with i.i.d. 0-mean subgaussian entries with parameter $1/d$, then we have*

$$\mathbb{P}(||Mx||_2^2 - ||x||_2^2 > \epsilon ||x||_2^2) < e^{-d\epsilon^2/8},$$
$$\mathbb{P}(||Mx||_2^2 - ||x||_2^2 < -\epsilon ||x||_2^2) < e^{-d\epsilon^2/8}$$

for any $\epsilon \in (0, 1)$.

Lemma 3.7.7 (Dot product under random projection) *Let $x, y \in \mathbb{R}^D$ two arbitrary vectors and $M \in \mathbb{R}^{d \times D}$ a random matrix with i.i.d. 0-mean subgaussian entries with parameter $1/d$, then we have*

$$\mathbb{P}\left(\langle Mx, My \rangle - \langle x, y \rangle > \epsilon ||x||_2 ||y||_2\right) < e^{-d\epsilon^2/8},$$

and

$$\mathbb{P}\left(\langle Mx, My \rangle - \langle x, y \rangle < -\epsilon ||x||_2 ||y||_2\right) < e^{-d\epsilon^2/8}$$

for any $\epsilon \in (0, 1)$.

Lemma 3.7.8 *We have that the event $\mathcal{E}_1$ defined in (3.14) holds with probability at least $1 - \delta$ if $d \geq 8 \log((N + T)/\delta)/\epsilon^2$.*

Proof: This is actually a union bound using JL lemma (Lemma 3.7.6) applied on all θ_i and x_t. More specifically, we have

$$\begin{aligned}
&\mathbb{P}\left(\bigcup_{i\in\mathcal{N}}\left(||\tilde{\theta}_i||_2^2 > (1+\epsilon)||\theta_i||_2^2\right) \cup \bigcup_{t\in\mathcal{T}}\left(||\tilde{x}_t||_2^2 > (1+\epsilon)||x_t||_2^2\right)\right)\\
\leq &\sum_{i\in\mathcal{N}}\mathbb{P}\left(||\tilde{\theta}_i||_2^2 > (1+\epsilon)||\theta_i||_2^2\right) + \sum_{t\in\mathcal{T}}\mathbb{P}\left(||\tilde{x}_t||_2^2 > (1+\epsilon)||x_t||_2^2\right)\\
< &(N+T)e^{-d\epsilon^2/8},
\end{aligned}$$

where the second inequality is by union bound, and the third inequality is from Lemma 3.7.6. So when $d \geq 8\log((N+T)/\delta)/\epsilon^2$, we have that $(N+T)e^{-d\epsilon^2/8} \leq \delta$. □

Lemma 3.7.9 *The event $\mathcal{E}_2$ defined in (3.15) holds with probability at least $1-\delta$ when $d \geq 8\log(TN/\delta)/\epsilon^2$.*

Proof: First, when $x_{t;>d_0}$ or $\theta_{i;>d_0}$ is equal to 0,

$$|\langle x_t, \theta_i\rangle - \langle \tilde{x}_t, \tilde{\theta}_i\rangle| = 0 \leq \epsilon||x_{t;>d_0}||_2||\theta_{i;>d_0}||_2$$

holds trivially. So we focus on the case that $x_{t;>d_0}$ and $\theta_{i;>d_0}$ are nonzero. In this case, the γ gap assumption implies that $||x_{t;>d_0}||_2||\theta_{i;>d_0}||_2 \leq \gamma|x'_{t;>d_0}\theta_{i;>d_0}|$. Then the proof follows almost identically as that in Lemma 3.7.8 except that we now apply the inner product error of random projection in Lemma 3.7.7. □

Lemma 3.7.10 *On event $\mathcal{E}_1, \mathcal{E}_2$, with probability at least $1-\delta$ we have*

$$||\tilde{\theta} - \check{\theta}_t||_{\bar{W}_t} \leq \bar{\alpha}_T = O\big(K^{3/2}(\sqrt{N(d_0+d)\log(T/\delta)} + \epsilon/\gamma\sqrt{LN})\big).$$

Proof: First of all, it is straightforward to check that results in Lemma 3.7.4 still hold (with $\bar{\beta}_t$ different only in constant factors) when θ is replaced by $\tilde{\theta}$ and x_t replaced by $\tilde{x}_t$ because $\tilde{\theta}_i\tilde{x}_t$ are still bounded on event $\mathcal{E}_1$. Then by Taylor's theorem, we have

$$\begin{aligned}
\bar{\beta}_t \geq &\sum_{s=1}^{t}(l_s(\bar{z}_s, Y_s) - l_s(\tilde{z}_s, Y_s)) \geq \sum_{s=1}^{t}\nabla l_s(\tilde{z}_s, Y_s)'(\bar{z}_s - \tilde{z}_s) + \sum_{s=1}^{t}\frac{1}{2\Gamma_s}||\bar{z}_s - \tilde{z}_s||_2^2\\
= &\sum_{s=1}^{t}\sum_{i\in S_s}\tilde{c}_{i,s}\tilde{x}'_s(\bar{\theta}_{i,s} - \tilde{\theta}_i) + \sum_{s=1}^{t}\frac{|S_s|}{2\Gamma_s}\sum_{i\in S_s}(\tilde{x}'_s(\bar{\theta}_{i,s} - \tilde{\theta}_i))^2/|S_s|,
\end{aligned} \tag{3.33}$$

where

$$\tilde{c}_{i,s} = \tilde{q}_{i,s}(\tilde{\theta}) - Y_{i,s} = \tilde{q}_{i,s}(\tilde{\theta}) - q_{i,s}(\theta) + q_{i,s}(\theta) - Y_{i,s} = \tilde{q}_{i,s}(\tilde{\theta}) - q_{i,s}(\theta) - \epsilon_{i,s}.$$

As a result, we have

$$\sum_{i\in S_s} \tilde{c}_{i,s}\tilde{x}'_s(\bar{\theta}_{i,s} - \tilde{\theta}_i) = \sum_{i\in S_s}(\tilde{q}_{i,s}(\tilde{\theta}) - q_{i,s}(\theta))\tilde{x}'_{i,s}(\bar{\theta}_s - \tilde{\theta}) - \sum_{i\in S_s}\epsilon_{i,s}\tilde{x}'_{i,s}(\bar{\theta}_s - \tilde{\theta}).$$

By a bit abuse of notation, we still let $\mu_{i,s} := \tilde{x}'_{i,s}(\bar{\theta}_s - \tilde{\theta})$. By Taylor's theorem, we can show that

$$\begin{aligned}
&\sum_{s=1}^{t}\sum_{i\in S_s}(\tilde{q}_{i,s}(\tilde{\theta}) - q_{i,s}(\theta))\tilde{x}'_{i,s}(\bar{\theta}_s - \tilde{\theta}) \\
=&-\sum_{s=1}^{t}\sum_{i\in S_s}\bar{q}_{i,s}(\mu_{i,s} - \sum_{j\in S_s}\bar{q}_{j,s})x'_{i,s}\theta + \sum_{s=1}^{t}\sum_{i\in S_s}\bar{q}_{i,s}(\mu_{i,s} - \sum_{j\in S_s}\bar{q}_{j,s}\mu_{j,s})\tilde{x}'_{i,s}\tilde{\theta} \\
=&(\tilde{B}'_{1,t}\tilde{\theta} - B'_{1,t}\theta) - (\tilde{B}'_{2,t}\tilde{\theta} - B'_{2,t}\theta),
\end{aligned}$$

where $\bar{q}_{i,s}$ is defined by finding a middle point between $\tilde{\theta}'_i\tilde{x}_{i,s}$ and $\theta'_i x_{i,s}$ for all i, s, and

$$B_{1,t} := \sum_{s=1}^{t}\sum_{i\in S_s}\bar{q}_{i,s}\mu_{i,s}x_{i,s}, \quad B_{2,t} := \sum_{s=1}^{t}\sum_{i\in S_s}\bar{q}_{i,s}x_{i,s}\sum_{j\in S_s}\bar{q}_{j,s}\mu_{j,s}.$$

Note that on event $\mathcal{E}_1$, $\bar{q}_{i,s} \le (\overline{\kappa}/\underline{\kappa})^{\sqrt{2}}/|S_s|$. So we have that

$$\begin{aligned}
|\tilde{B}'_{1,t}\tilde{\theta} - B'_{1,t}\theta| \le& \left(\frac{\overline{\kappa}}{\underline{\kappa}}\right)^{\sqrt{2}}\sum_{s=1}^{t-1}\sum_{i\in S_s}\frac{|\mu_{i,s}|}{\sqrt{|S_s|}}\frac{|x'_{i,s}\theta - \tilde{x}'_{i,s}\tilde{\theta}|}{\sqrt{|S_s|}} \\
\le& \left(\frac{\overline{\kappa}}{\underline{\kappa}}\right)^{\sqrt{2}}\sqrt{\sum_{s=1}^{t-1}\sum_{i\in S_s}\frac{\mu_{i,s}^2}{|S_s|}}\sqrt{\sum_{s=1}^{t-1}\sum_{i\in S_s}\frac{(x'_{i,s}\theta - \tilde{x}'_{i,s}\tilde{\theta})^2}{|S_s|}} \\
\le& \left(\frac{\overline{\kappa}}{\underline{\kappa}}\right)^{\sqrt{2}}\sqrt{\sum_{s=1}^{t-1}\sum_{i\in S_s}\frac{\mu_{i,s}^2}{|S_s|}}\sqrt{\epsilon^2/\gamma^2\sum_{i\in\mathcal{N}}\sum_{t\in\mathcal{T}_i}||\theta_{i;B_t}||_2^2} \\
\le& \frac{\epsilon\overline{\kappa}^{\sqrt{2}}R}{\underline{\kappa}^{\sqrt{2}}\gamma}\sqrt{LN}\sqrt{\sum_{s=1}^{t-1}\sum_{i\in S_s}\frac{\mu_{i,s}^2}{|S_s|}},
\end{aligned}$$

where the second inequality is by Cauchy-Schwarz, the third inequality is from the definition of $\mathcal{E}_2$, and the last inequality is from the sparsity assumption. We can similarly

show

$$|\tilde{B}'_{2,t}\tilde{\theta} - B'_{2,t}\theta| \leq \frac{\epsilon\overline{\kappa}^{\sqrt{2}}R}{2\underline{\kappa}^{\sqrt{2}}\gamma}\sqrt{LN}\sqrt{\sum_{s=1}^{t}\sum_{i\in S_s}\mu_{i,s}^2/|S_s|}.$$

Thus, we obtain that on $\mathcal{E}_1, \mathcal{E}_2$, for all $t \in \mathcal{T}$,

$$\left|\sum_{s=1}^{t}\sum_{i\in S_s}(\tilde{q}_{i,s}(\tilde{\theta}) - q_{i,s}(\theta))\tilde{x}'_{i,s}(\bar{\theta}_s - \tilde{\theta})\right| \leq \overline{\kappa}^{\sqrt{2}}\epsilon R/(\underline{\kappa}^{\sqrt{2}}\gamma)\sqrt{LN\sum_{s=1}^{t}\sum_{i\in S_s}\mu_{i,s}^2/|S_s|}.$$

The analysis of $\sum_{s=1}^{t}\sum_{i\in S_s}\epsilon_{i,s}\tilde{x}'_{i,s}(\bar{\theta}_s - \tilde{\theta})$, is the same as that in Lemma 3.7.5 except that, everything is in the projected space. We omit the details for brevity. As a result, like what we did in Lemma 3.7.5, we have that

$$\begin{aligned}\sum_{s=1}^{t}\sum_{i\in S_s}\mu_{i,s}^2/|S_s| \leq & k'_9K^{3/2}\sqrt{2\left(1+\sum_{s=1}^{t}\sum_{i\in S_s}\mu_{i,s}^2/|S_s|\right)\log\left(\sqrt{1+\sum_{s=1}^{t}\sum_{i\in S_s}\mu_{i,s}^2/|S_s|}/\delta\right)} \\ & + k'_9K\bar{\beta}_t + 2\overline{\kappa}^{\sqrt{2}}\epsilon R/(\underline{\kappa}^{\sqrt{2}}\gamma)K\sqrt{LN\sum_{s=1}^{t}\sum_{i\in S_s}\mu_{i,s}^2/|S_s|}\end{aligned}$$

for some constant k'_9. Using Lemma 2 in Jun et al. (2017), we have that

$$\sum_{s=1}^{t}\sum_{i\in S_s}\mu_{i,s}^2/|S_s| \leq k'_{10}K\bar{\beta}_t + k'_{10}K^3\log(T/\delta) + k'_{10}K^2\epsilon^2LN/\gamma^2$$

for some constant k'_{10}. Then the rest of the proof just follows the steps after equation (3.32) in Lemma 3.7.5.

□

3.8 Conclusion

With the rapid development of information technology, mass customization is becoming increasingly popular in online retailing and online advertising, and personalized assortment is one of the most important decisions for customization. In this chapter, we study an online personalized assortment optimization problem, where customers' preferences toward products are not known *a priori*. To model customers' preferences, we adopt the widely accepted multinomial logit (MNL) model with unknown choice parameters. We design

two adaptive algorithms that learn the demand on the fly. The first one, P-UCB, uses MLE for parameter estimation and applies personalized UCB for assortment optimization in demand exploration. We prove that the regret of P-UCB is at most $\tilde{O}(DNK\sqrt{T})$. The second algorithm, OLP-UCB, bears similar structure as P-UCB but applies an online convex optimization scheme for parameter optimization. OLP-UCB has a constant computational time (in contrast to linearly increasing time of P-UCB) in each iteration, so it significantly reduces computational cost when large historical data has been collected. With drastic improvement in computational efficiency, we show that the OLP-UCB algorithm achieves a regret of $\tilde{O}(DNK^{3/2}\sqrt{T})$. We then consider the online personalized assortment optimization problem with high dimensional customers' data. Motivated by our observation of industry data that customers' information has sparse structure, we apply random projection method to tackle the high dimensionality challenge, which significantly reduces dimensionality and computational cost. The OLP-UCB-RP algorithm developed for high dimensional problem achieves regret rate $\tilde{O}(NK^{3/2}\sqrt{(d_0+d)LT}+(d_0+d)NK^{3/2}\sqrt{T})$, where d_0, L are parameters related to the data sparsity, and d (which is much smaller than D) is the dimension of data after random projection.

There are several potential future research directions of this work. First, as we have shown in Theorem 3.3.2, there are some gaps between the regret upper bound of our algorithms and the regret lower bound. So closing these gaps will be a technical contribution for online personalized assortment optimization problem. Another possible direction to explore is alternative dimension reduction techniques in regret minimization. For example, besides random projection method, principal component analysis (PCA) has been widely used for dimension reduction. Analyzing the performance of learning algorithms with other dimension reduction method such as PCA is an interesting research problem. Finally, in this chapter we consider the case when customer choice follows an MNL model. Personalized assortment optimization problem for other, and more general, choice models is worthy of investigation.

Chapter 4

Dynamic Joint Assortment and Pricing Optimization with Demand Learning

4.1 Introduction

It is common in retailing, e-commerce and advertisement settings, that customers or users are presented with a set of products simultaneously, known as an assortment, to induce them to purchase. Due to system capacity (e.g., limited shelf space, budget constraint), the firm can only display up to a certain number of products at a time. Thus, the firm has to decide which assortment to offer at any time in order to maximize a certain objective function (e.g., the number of clicks, expected revenue or profit). This is known as assortment optimization. The assortment optimization problem has become an active research area in the operations literature in recent years, especially with the increasing popularity of online shopping. Equally important in retailing is dynamic pricing. Pricing enables firms to increase revenue by better matching supply with demand and by responding quickly to a demand pattern, and it also has attracted enormous research interest in the revenue management literature. We refer the reader to the comprehensive survey papers of Kök et al. (2015) for assortment optimization and of Yano and Gilbert (2005) and Chen and Chen (2015) for dynamic pricing.

Although both assortment optimization and dynamic pricing problems have been extensively studied alone in the operations literature, there are relatively few papers on joint assortment and pricing optimization. At the end of their survey paper, Kök et al. (2015) state that "the joint pricing and assortment planning problem has not been studied in depth" and list that as a future research direction. For some interesting studies on the joint optimization of assortment planning and pricing, see Wang (2012), Jagabathula and Rusmevichientong (2015) and the references therein (see Section 4.1.2 for more details).

Until recently, the majority of the existing literature on assortment optimization and/or pricing are based on the assumption that the firm has exact prior knowledge about the

demand distribution and customer choice behavior. This may not be true in many applications. For example, in the fast fashion industry, the random demand of products is affected by both assortment and selling prices, but the extent to which the demand is affected may not be precisely known at the start. Hence, it is important for the firm to understand how customers make choices in order to *dynamically set assortment and pricing* decisions via *demand learning* to maximize revenue.

In this chapter, we study the dynamic joint assortment optimization and pricing problem when customers follow the multinomial logit (MNL) choice model, but the choice parameters are not known to the firm *a priori*. We choose the MNL model not only for its computational tractability (see e.g., Ryzin and Mahajan 1999, Rusmevichientong et al. 2010, Wang 2012), but also for its practical significance (see Feldman et al. 2018 for a recent field experiment result of assortment optimization using the MNL choice model). Specifically, we consider a firm that sells N products over a planning horizon of T periods, where T may or may not be known at the beginning. The firm can display up to K products in an assortment at any time. There is exactly one arrival in each period that either buys a product from the display, or leaves without any purchase. Therefore, a period can also be considered as an arriving customer, and then T is interpreted as the total number of arrivals in question. The objective of the firm is to maximize the expected total revenue over the planning horizon. Not knowing the choice parameters in advance, the firm needs to learn the demand information on the fly.

Intuitively, the firm has to spend sufficient time testing on each product in order to adequately understand the customer's taste to it. On the other hand, due to the finite planning horizon, the more time the firm spends on demand learning (exploration), the less time will be left for exploitation to extract revenue. Hence, the essence in achieving effective demand learning is to strike a balance between exploration and exploitation. In this chapter we design a learning algorithm that balances the trade-off between demand learning and revenue extraction, and evaluate the performance of the algorithm using Bayesian regret, which is the average (expected) revenue loss compared with a clairvoyant that has complete information about customer choice parameters. We derive a theoretical upper bound for the Bayesian regret of our algorithm, and it is independent of the specific choice parameters of the model. Numerical experiments are also conducted and the results show that our algorithm outperforms the benchmarks. For brevity (and with some abuse of terms), in this chapter if an algorithm has *instance-independent performance upper bound*, we will say that this algorithm has *instance-independent* performance; otherwise, it has instance-dependent performance.

4.1.1 Main contributions of the chapter

The main contributions of this chapter are summarized as follows.

a) We present the first learning algorithm for dynamic joint assortment optimization and pricing problem when customer choice follows the MNL model but choice parameters are not known *a priori*. The algorithm exploits the structure of the MNL model using a concept of *cycles* introduced in Agrawal et al. (2017a), and it divides the cycles into three distinct categories that are optimally designed for balancing exploration and exploitation. The algorithm further incorporates Thompson Sampling (see e.g., Russo and Van Roy 2014) in the appropriately defined, so-called, "sufficiently tested and priced" cycles. We evaluate the algorithm using Bayesian regret, and prove that it has an *instance-independent* Bayesian regret upper bound $O\big(N \log(NT) + \sqrt{NT}\log(NT)\big)$ (Theorem 4.3.1). Numerical experiments show that the algorithm exhibits excellent performance in terms of regret. In particular, the algorithm is compared against several benchmarks, including a popular epsilon-First algorithm (exploration then exploitation), a modified epsilon-First algorithm, and a natural alternative of our algorithm based on parameter estimation. Our algorithm outperforms the benchmarks in all numerical experiments.

b) We establish a concentration inequality for none i.i.d. sub-exponential random errors, with stochastic contexts (Proposition 4.5.2). This concentration inequality plays a pivotal role in establishing an important confidence bound of maximum likelihood estimator which later leads to the Bayesian regret upper bound of our algorithm. The concentration inequality extends a result of Abbasi-Yadkori et al. (2011) for sub-Gaussian random errors (a stronger assumption than sub-exponential), and it is expected to be useful in conducting regret analysis for other settings.

4.1.2 Related literature

We next review the related literature. Besides the literature on assortment optimization and dynamic pricing, we will also discuss the relevant works in the area of *multi-armed bandit* (MAB) problems. Following the literature, we use $O(\cdot)$ to denote the regret upper bound, and use $\Omega(\cdot)$ to denote the regret lower bound. We also use $\tilde{O}(\cdot)$ to represent the upper bound which hides the logarithmic terms.

Static and dynamic assortment optimization. Assortment optimization has drawn much attention in the operations literature especially in recent years. Traditional research of assortment optimization focuses mainly on the static problem where the firm knows

demand information *a priori*. Kök et al. (2015) is an extensive survey on static assortment optimization problems. Thus here we only highlight some representative work. Ryzin and Mahajan (1999) study a single period assortment optimization problem with the MNL choice model. Assuming that the sale is lost whenever the customer finds that the product he or she chooses is out of stock, they prove that the optimal assortment consists of a certain number of the most popular products. When inventory planning is incorporated, Goyal et al. (2016) prove that the static joint assortment and inventory planning problem under dynamic substitution is NP-hard, and they present a polynomial-time approximation algorithm. See Hopp and Xu (2008), Honhon et al. (2010) for analysis of similar problems.

Dynamic assortment optimization is another important class of revenue management problem (see e.g., Aouad et al. 2018), and in particular, the problem with demand learning has attracted increasing interest in recent years. To the best of our knowledge, Caro and Gallien (2007) are the first to study the dynamic assortment optimization problem. The authors assume that the demand of each product is independent, and a Bayesian updating is used to learn the demand information of each product. Rusmevichientong et al. (2010) consider both static and dynamic optimization. Adopting the classical MNL choice model for the demand but with unknown choice parameters, the authors propose a learning algorithm with regret $O(N^2 \log T)$ which depends on the problem instance (in particular, it depends on the minimum gap between the optimal revenue and the revenue of any sub-optimal assortment). A more general model is proposed in Sauré and Zeevi (2013) for which the MNL choice model is a special case. Their algorithm has instance-dependent regret $O(N/K \log T)$, which matches the instance-dependent lower bound $\Omega(N/K \log T)$ established by the same authors. The dynamic assortment optimization using the MNL choice model is further analyzed in Agrawal et al. (2017a), where the special structure of the MNL model is exploited and an algorithm is developed based on the upper confidence bound (UCB) algorithm from the multi-armed bandit literature (see e.g., Auer et al. 2002). The performance of this algorithm does not depend on the problem instance, and it has an instance-independent regret bound $O(\sqrt{NT} \log T + N \log^3 T)$ which almost matches the instance-independent lower bound $\Omega(\sqrt{NT})$ (given $K \leq N/4$) proved in Chen and Wang (2017). Another algorithm for the MNL choice model based on Thompson Sampling is proposed in Agrawal et al. (2017b) with instance-independent regret $O(\sqrt{NT} \log(TK) + N \log^2(TK))$. Cheung and Simchi-Levi (2017) use Thompson Sampling algorithm to solve the personalized dynamic assortment optimization problem. Assuming that the utility of each product depends on some time-varying contexts, the authors show that their algorithm achieves an instance-independent Bayesian regret $\tilde{O}(DN\sqrt{KT})$ where D is the dimension of the context, and $\tilde{O}(\cdot)$ hides logarithmic terms

of K, N, T. Selling prices are exogenous in these references.

Dynamic and multi-product pricing. Pricing is another important area of research in the operations literature and we refer the reader to Chen and Chen (2015) for a recent survey on it. We first review the literature on dynamic pricing with demand learning. Most of the works in the pricing literature with demand learning consider the single-product problem over the time horizon T. For instance, Broder and Rusmevichientong (2012) assume the demand function is a parametric model with unknown parameters, and they propose algorithms based on maximum likelihood estimation with regret $O(\sqrt{T})$, matching with the lower bound $\Omega(\sqrt{T})$. Furthermore, if the class of demand functions satisfy some "well-separated" condition, they prove that the regret can be reduced to $O(\log T)$, matching with lower bound $\Omega(\log T)$. Later on, den Boer and Zwart (2013) also use a parametric model and show that near-optimal regret rate can be achieved by controlling the variance of historical prices (via choosing prices from some carefully designed intervals). Keskin and Zeevi (2014) generalize this result and provide some sufficient conditions on price variance control. Based on these conditions, they propose a class of pricing policies called *semi-myopic pricing policies* which can achieve near-optimal regret rates. Besbes and Zeevi (2015) investigate the impact of model misspecification, and show that under certain conditions, linear demand approximation achieves near-optimal regret. For multi-product dynamic pricing, one of the first papers is Gallego and Van Ryzin (1997). Assuming that there are initial inventory of some common resources, and replenishment of resources is not allowed, Gallego and Van Ryzin (1997) propose two heuristics to solve the multi-product dynamic pricing problem, which are asymptotically optimal. Much follow-up work has been done when the underlying demand process is not known *a priori*. For instance, assuming that the demand follows a Poisson process with unknown arrival intensity, Besbes and Zeevi (2012) propose a data-driven learning algorithm with regret $O(\sqrt[3]{T^2}\sqrt{\log T})$, where T represents both the length of planning horizon and scale of initial inventory. Later, Ferreira et al. (2018b) consider a similar problem but with arbitrary demand function and discrete prices. They use Thompson Sampling and linear program to design a learning algorithm with Bayesian regret $O(\sqrt{TK \log K})$, where K is the number of feasible prices. If the inventory is infinite, den Boer (2014) proposes an algorithm with regret $O(\sqrt{T \log T})$ in the case of generalized linear demand with canonical link function.

In the operations and marketing literature, MNL is a well-accepted choice model for multi-product pricing. Hanson and Martin (1996) consider the static multi-product pricing problem with demand following the MNL model. They show that although the revenue function is not concave with respect to prices, the global optimum can be computed efficiently via a different convex optimization problem. A generalization of the MNL model

is the so-called nested logit (NL) model, and Li and Huh (2011) consider the multiproduct pricing with NL choice model. Both papers assume that the price sensitivities of different products are identical, which simplifies their analysis. Dong et al. (2009) study a pricing problem of multiple products, and Akday et al. (2010) study pricing and inventory control of multiple products; both of these papers use the MNL model but do not have assortment decisions. Gallego and Wang (2014) extend the problem to the NL model with different price sensitivities and report structural results. They show that the high dimensional pricing problem can be reduced to a one-dimensional problem, since the optimal prices of all products can be given in terms of a single decision variable.

Joint assortment optimization and pricing. Assortment optimization and pricing optimization have been studied separately extensively in the literature, but there are relatively few papers on the joint optimization problem. Among these, Wang (2012) has almost the same model as ours except that the demand is known *a priori*. Wang (2012) shows that the optimization problem with complete demand information exhibits an interesting special structure that allows for efficient computational algorithm for the optimal assortment and pricing decisions (using a certain bisection search method). Besbes and Sauré (2016) also use the MNL choice model for joint assortment and pricing optimization, but in a competitive environment. They show that there always exists a pure-strategy Nash equilibrium when different retailers offer non-overlapping products to customers. When products of different retailers have overlap, they show that a pure-strategy Nash equilibrium exists if there is no display capacity constraint. Other papers in the literature on joint assortment optimization and pricing problem include Chen and Hausman (2000), Kök and Xu (2011), Rodríguez and Aydın (2011), Jagabathula and Rusmevichientong (2015), Alptekinoğlu and Semple (2016), Chen and Jiang (2017), and others.

To the best of our knowledge, there is only one paper (Talebian et al. 2012) which considers the joint optimization of assortment optimization and pricing when demand information is not known *a priori*. However, Talebian et al. (2012) assume that the demand of each product is independent, which does not hold under the MNL choice model considered in this chapter. Our problem cannot be effectively solved by combining learning algorithms for dynamic assortment optimization and pricing problems. Indeed, the optimal assortments under different product prices are different, and the optimal prices of the same product in different assortments are also different. Pure multi-product pricing algorithms (e.g., den Boer 2014) cannot be applied to our problem either. One might hope to apply the pure multi-product pricing algorithm to our problem by setting the price of certain product extremely high, which can be viewed as not selecting this product in the assortment. This, however, leads to two issues. First, multi-product pricing algorithms (e.g., den Boer 2014)

require the prices to be chosen within a compact set, and their regrets depend critically on the upper bound of the prices. So if the upper bound of the prices is very high, it will lead to pretty bad regret. Second, the key for any assortment optimization problem is to select a set of products to offer, which is combinatorial in nature, and it is not clear how to solve a multi-product pricing algorithm with a constraint on the number of products (i.e., how to use the existing algorithms for multiple-product problem with a constraint on the number of products).

Related work in multi-armed bandit (MAB) problem. The MAB problem is a classic research area that is attracting growing interest in recent years due to its broad application in machine learning. Many of the online learning problems in operations can be formulated as MAB. For a comprehensive literature review and introduction to available algorithms for MAB, we refer the reader to Bubeck et al. (2012), Lattimore and Szepesvári (2018). In fact, both assortment and pricing optimization problems can be formulated as MAB. For instance, each assortment can be considered as an "arm" in the MAB setting, and classical algorithms of discrete MAB can be directly applied (Bubeck et al. 2012). However, direct application leads to the curse of dimensionality because the resulting MAB problem has $\binom{N}{K}$ arms where N is the total number of products, and K is the assortment capacity. For multiproduct dynamic pricing, it can be considered as an online optimization with bandit feedback, although the reward function (which is the revenue function in our setting) is not concave with respect to prices even under the MNL choice model (Hanson and Martin 1996). Online convex optimization problem with bandit feedback is one of the most important problems in MAB research, and an algorithm with regret $O(\sqrt{T})$ and polynomial of space dimension has been recently derived in Bubeck et al. (2016).

4.1.3 Organization of the chapter

The remainder of this chapter is organized as follows. In Section 4.2 we introduce the multinomial logit model and other model specifics. A Thompson Sampling based learning algorithm, termed TS-PS, is developed in Section 3 and it is shown that the algorithm has instance-independent Bayesian regret upper bound. Numerical experiments of the algorithm and several benchmark algorithms are conducted in Section 4.4. The sketches for the proofs of the main results are given in Section 5, but we leave some of the technical details in the section of proofs. Finally, we conclude the chapter in Section 4.7.

4.2 Problem Formulation

A firm has N products to sell over T periods, where T may be unknown *a priori*. The products are labeled as $i = 1, 2, \ldots, N$, and periods labeled by $t = 1, 2, \ldots, T$. In each period, the firm can display up to K products, called an assortment. During each period, there is exactly one arrival that either purchases one of the products on display or leaves without purchasing any product. We consider the scenario that the firm is able to replenish its stock quickly and that the assortment can be changed with negligible time/cost. The firm needs to determine, for each period, the set of products to display (assortment decision) and the selling prices of the displayed products (pricing decision), and its objective is to maximize the expected total revenue over the planning horizon.

For convenience, we denote the set of all products by $\mathcal{N} := \{1, 2, \ldots, N\}$, and the collection of all possible assortments by $\mathcal{S} := \{S \subset \mathcal{N} : \ |S| \leq K\}$. Here, and in what follows, ":=" stands for "defined as", and $|S|$ denotes the number of elements in, or the cardinality of, assortment S.

We adopt the popular multinomial logit (MNL) model for customer choice. It is well-known that the MNL choice model is obtained from a common Gumbel utility value distribution. See, e.g., Ben-Akiva and Lerman (1985). Under the MNL model, when an assortment S is offered at prices $p := (p_i : \ i \in \mathcal{N}) \in \mathbb{R}^N$ (with this definition of price vector, customers only observe price p_i for $i \in S$ and prices of the other products are not observable), the probability that an arriving customer purchases product $i \in S \cup \{0\}$, with 0 representing the no-purchase option, is

$$\begin{aligned} q(i|S,p) :=& \frac{v_i(p_i)}{v_0 + \sum_{j \in S} v_j(p_j)}, \quad i \in S, \\ q(0|S,p) :=& \frac{v_0}{v_0 + \sum_{j \in S} v_j(p_j)}, \end{aligned} \tag{4.1}$$

where $v_i(p_i) := e^{\alpha_i - \beta_i p_i}$ and $v_0 := e^{\alpha_0}$, and the customer choice parameters α_i and β_i represent the feature utility and price sensitivity of product i, respectively. Without loss of generality (by dividing v_0 on both nominator and denominator of each $q(i|S,p)$), we assume that $\alpha_0 = 0$ or $v_0 = 1$. For ease of notation, we use $\theta_i' = (\alpha_i, \beta_i)$ to denote the choice parameter of product i, where θ_i' is the transpose of column vector θ_i, and use $\theta' = (\theta_1', \ldots, \theta_N')$ to represent the parameters of all products. The parameters α_i and β_i take values in $[\underline{\alpha}_i, \overline{\alpha}_i]$ and $[\underline{\beta}_i, \overline{\beta}_i]$, respectively, with $\underline{\beta}_i > 0$. This assumption is plausible as it basically states that, everything else being equal, a customer prefers lower price than higher price.

For convenience we denote the feasible set of parameters of θ_i as Θ_i, the feasible set of θ as $\Theta = \bigotimes_{i\in\mathcal{N}} \Theta_i$. Furthermore, we assume that the selling price p_i is in the range $\mathcal{P}_i := [\underline{p}_i, \overline{p}_i]$ for some $0 \le \underline{p}_i < \overline{p}_i < \infty$. Let $\mathcal{P} = \bigotimes_{i\in\mathcal{N}} \mathcal{P}_i$ denote the region of feasible selling prices. For ease of presentation, we let

$$\underline{\alpha} := \min_{i\in\mathcal{N}} \underline{\alpha}_i, \quad \underline{\beta} := \min_{i\in\mathcal{N}} \underline{\beta}_i, \quad \overline{\alpha} := \max_{i\in\mathcal{N}} \overline{\alpha}_i, \quad \overline{\beta} := \max_{i\in\mathcal{N}} \overline{\beta}_i, \quad \underline{p} := \min_{i\in\mathcal{N}} \underline{p}_i, \quad \overline{p} := \max_{i\in\mathcal{N}} \overline{p}_i. \tag{4.2}$$

In this chapter, we call a number *constant* if it depends only on the range of possible parameters Θ and $\mathcal{P}$, and not on the specific values of the problem parameters.

Given an assortment S and selling prices p, the expected revenue from an arriving customer, denoted by $r(S, p, \theta)$, is

$$r(S, p, \theta) := \sum_{i\in S} p_i \cdot q(i|S, p). \tag{4.3}$$

We include θ here because the revenue function depends on customer choice parameters. However, we often omit θ when no confusions may arise, i.e., simply write it as $r(S, p)$.

Clairvoyant problem. If all the choice parameters θ_i, $i = 1, \ldots, N$, are known *a priori*, then the revenue function (4.3) can be maximized to find an optimal solution S^* and price $p^* = (p_i^* : i \in \mathcal{N})$, i.e.,

$$(S^*, p^*) \in \arg\max_{S\in\mathcal{S}, p\in\mathcal{P}} r(S, p). \tag{4.4}$$

We refer to (S^*, p^*) as a clairvoyant solution. To emphasize its dependency on parameters θ, we often write it as $(S^*(\theta), p^*(\theta))$. The clairvoyant maximum total revenue over the planning horizon is

$$J_\theta^*(T) := \sum_{t=1}^{T} r(S^*, p^*) = T \cdot r(S^*, p^*),$$

and it will be used as a benchmark for analyzing the performance of our learning algorithm. Note that the optimal assortment and pricing problem (4.4) with known customer choice parameters has been studied in the literature, and efficient computational algorithms are available to find an optimal solution (see e.g., Wang 2012).

In many applications, it is unlikely that all parameters α_i and β_i are known to the firm *a priori* but nonetheless, the firm has to make assortment and pricing decisions in every period. Then, how to design a learning mechanism to extract revenue that is as close to that of the clairvoyant solution as possible? This is the problem we study in this chapter.

The firm's optimization problem. Not knowing the customer preference parameters

of the choice model *a priori* and subjecting to display capacity constraint, the firm's problem is to determine a dynamic policy π which sets the display S_t and their selling prices $p_t = (p_{i,t} : \ i \in \mathcal{N})$ for each period t, based on information up to $t-1$, for $t = 1, 2, \ldots, T$, so that the expected total revenue

$$J_\theta^\pi(T) := \mathbb{E}\Big[\sum_{t=1}^{T} r(S_t, p_t) \ \Big| \ \theta \Big]$$

is as large as possible, where the expectation is taken over the randomness introduced by π and customer's choice. The policy π is nonanticipative in that the choice of (S_t, p_t) can depend only on the history $\mathcal{F}_{t-1}$ which is defined as the σ-algebra generated by $\{(S_s, p_s, i_s) : s = 1, \ldots, t-1\}$, where $i_s \in S_s \cup \{0\}$ is customer's choice at time s.

Bayesian regret. Before introducing Bayesian regret, we first note that another commonly used metric to measure the effectiveness of a learning algorithm π is regret, see e.g., Bubeck et al. (2012), defined as the expected revenue loss of the algorithm π compared with a clairvoyant optimal solution. That is,

$$R_\theta^\pi(T) := J_\theta^*(T) - J_\theta^\pi(T) = \mathbb{E}\Big[\sum_{t=1}^{T} (r(S^*, p^*) - r(S_t, p_t)) \ \Big| \ \theta \Big].$$

For Bayesian regret, it is assumed that the unknown parameter θ is randomly drawn from its domain Θ according to a prior distribution Φ_1 (which is arbitrary but is known to the firm, for the numerical experiment in Section 4.4, we use uniform distribution for illustration). Thus the Bayesian regret is the expected regret, defined as

$$BR^\pi := \mathbb{E}_\theta[R_\theta^\pi(T)],$$

where the expectation $\mathbb{E}_\theta[\cdot]$ is taken with respect to the prior distribution Φ_1 over θ. Therefore, regret can be considered as the performance of the algorithm given (a realization of) θ, while Bayesian regret is the average regret over all θ from its prior distribution Φ_1. Bayesian regret is a weaker notion than regret, because if the regret can be bounded above by some number for all θ in Θ, then the expected regret is also upper bounded by the same number. For more discussion on Bayesian regret and its relationship with regret, we refer the interested readers to Russo and Van Roy (2014).

Clearly, the smaller the Bayesian regret, the better the learning algorithm performs. Our goal in this chapter is to design an adaptive algorithm that learns the customer choice behavior on the fly, and that the rate of its Bayesian regret is as small as possible.

We end this section by pointing out that, in principle it is possible to solve this problem

using Markov decision process (MDP) through Bayesian updating. However, the MDP approach suffers from curse of dimensionality, hence Thompson Sampling is adopted in our online learning scheme of this chapter.

4.3 Algorithm and Main Result

In this section, we present a learning algorithm with instance-independent Bayesian regret upper bound for the dynamic joint assortment and pricing optimization problem. We describe the detailed algorithm in Section 4.3.1, and then we present the theoretical result on the performance of the algorithm, followed by a discussion on intuitions and insights, in Section 4.3.2.

4.3.1 Algorithm description

Our algorithm design is inspired by a cycle-based approach for pure dynamic assortment optimization problem in Agrawal et al. (2017a), and we integrate it with an online learning procedure known as Thompson Sampling (or posterior sampling, see e.g., Russo and Van Roy 2014). Before presenting the algorithm, we first introduce some important concepts.

Cycle approach. The idea of cycle was developed in Agrawal et al. (2017a) for a pure assortment optimization problem. Using cycle approach, the time horizon $\mathcal{T}$ is divided into cycles indexed by $l = 1, 2, \ldots$.

A cycle is defined as the set of time periods that we repeatedly offer an assortment with given prices until a no-purchase outcome occurs. More specifically, in a cycle l, we offer an assortment S_l with prices p_l repeatedly until a no-purchase happens in response to offering S_l, including the time period in which there is no purchase. Therefore each cycle, except the last one of the planning horizon, ends with a period of no-purchase. Let E_l denote the set of time periods in cycle l, and t_l denotes the first time period of cycle l. In each cycle l, we record the number of customers who purchase product $i \in S_l$ by $\hat{v}_{i,l}$, i.e.,

$$\hat{v}_{i,l} := \sum_{t \in E_l} \mathbf{1}(i_t = i),$$

where i_t is the product (including the no-purchase) that customer chooses to purchase at time t, and $\mathbf{1}(\mathcal{A})$ denotes the indicator function that takes value 1 if statement $\mathcal{A}$ is true and 0 otherwise. This approach decomposes the T periods into L_T cycles, with the last cycle possibly being incomplete. Note that L_T is a random variable and mathematically,

L_T is the cycle l that satisfies $i_t \neq 0$ for $t = t_l, t_l + 1, \ldots, T$ if $i_T \neq 0$, and $i_t \neq 0$ for $t = t_l, t_l + 1, \ldots T - 1$ if $i_T = 0$.

An important property of this cycle approach is that the marginal distribution of $\hat{v}_{i,l}$ depends only on the price $p_{i,l}$ of product i itself, and not on any other products (Agrawal et al. 2017a; also see Lemma 4.3.1). To emphasize this, we often write $\hat{v}_{i,l}$ as $\hat{v}_{i,l} = \hat{v}_{i,l}(p_{i,l})$. This property will be used later in establishing the performance bound of our algorithm.

Posterior sampling of θ. As in a typical Thompson Sampling algorithm, in a cycle l, we sample the unknown parameter $\tilde{\theta}_l$ according to a posterior distribution Φ_l, which satisfies

$$\Phi_l(\theta) \propto \Phi_1(\theta)\mathcal{L}(\theta|\mathcal{F}_{l-1}), \tag{4.5}$$

where "$\propto$" stands for "proportional", $\mathcal{F}_{l-1}$ is the history until the end of cycle $l-1$, and $\mathcal{L}(\theta|\mathcal{F}_{l-1})$ is the (joint) likelihood function given history $\mathcal{F}_{l-1}$. Therefore, besides depending on the prior distribution, Φ_l also depends on the likelihood function and the history $\mathcal{F}_{l-1}$. To illustrate, we present two examples.

Example 1. If we let $\mathcal{F}_{l-1}$ be the σ-algebra generated by (S_t, p_t, i_t) for $t = 1, \ldots, t_l - 1$, then the likelihood function can be defined as

$$\mathcal{L}(\theta|\mathcal{F}_{l-1}) = \prod_{t=1}^{t_l-1} q(i_t|S_t, p_t, \theta), \tag{4.6}$$

where $q(i_t|S_t, p_t, \theta)$, given in (4.1), is the purchasing probability of product i_t, given parameter θ, when assortment S_t and selling prices p_t are offered.

Example 2. If we only record the number of purchases of product i as $\hat{v}_{i,\tau}(p_{i,\tau})$ in each cycle τ, and let $\mathcal{F}_{l-1}$ be the σ-algebra generated by $(S_\tau, p_\tau, \{\hat{v}_{i,\tau}(p_{i,\tau})\}_{i\in S_\tau})$ for $\tau \leq l-1$, then the likelihood function will be different from the previous example because we do not have any information about i_t. In this case, the likelihood function is defined as

$$\mathcal{L}(\theta|\mathcal{F}_{l-1}) = \prod_{\tau=1}^{l-1} \mathbb{P}\big(\{\hat{v}_{i,\tau}(p_{i,\tau})\}_{i\in S_\tau}|S_\tau, p_\tau, \theta\big),$$

where the probability $\mathbb{P}\big(\{\hat{v}_{i,\tau}(p_{i,\tau})\}_{i\in S_\tau}|S_\tau, p_\tau, \theta\big)$ (which depends on the realization of $\{\hat{v}_{i,\tau}(p_{i,\tau})\}_{i\in S_\tau}$) can be derived as

$$\mathbb{P}\big(\{\hat{v}_{i,\tau}(p_{i,\tau})\}_{i\in S_\tau}|S_\tau, p_\tau, \theta\big) = \frac{(\sum_{i\in S_\tau} \hat{v}_{i,\tau}(p_{i,\tau}))!}{\prod_{i\in S_\tau} \hat{v}_{i,\tau}(p_{i,\tau})!} \left(\prod_{i\in S_\tau} q(i|S_\tau, p_\tau, \theta)^{\hat{v}_{i,\tau}(p_{i,\tau})}\right) q(0|S_\tau, p_\tau, \theta).$$

This is because, for each cycle τ, the realized data is $\{\hat{v}_{i,\tau}(p_{i,\tau})\}_{i\in S_\tau}$, and its probability

follows the standard multinomial distribution.

Random price shock. An important mechanism we utilize in learning customer choice parameters is price shock. That is, for some cycles, we add a random price perturbation to the selling price of an offered product. Essentially, in cycle l, we first calculate an "optimal" price $\hat{p}_l$ (for assortment S_l chosen by the algorithm in cycle l, see Algorithm 6) derived according to the sampled parameter $\tilde{\theta}_l$, i.e.,

$$\hat{p}_l \in \arg\max_{p \in \mathcal{P}} r(S_l, p, \tilde{\theta}_l).$$

Then, if it is determined (using the criterion in the algorithm) that some product $i \in S_l$ requires further "price exploration", then we set its selling price to $p_{i,l} = p'_{i,l} + \omega_{i,l}$, where $p'_{i,l} := \text{Proj}_{[\underline{p}_i + \omega_0, \overline{p}_i - \omega_0]}(\hat{p}_{i,l})$ with $\text{Proj}_{[a,b]}(x) := \max(\min(x, b), a)$, that maps $\hat{p}_{i,l}$ to the range $[\underline{p}_i + \omega_0, \overline{p}_i - \omega_0]$ in case it falls outside of it, ω_0 is a parameter satisfying $0 < \omega_0 \leq \min_{i \in \mathcal{N}}(\overline{p}_i - \underline{p}_i)/2$, and $\omega_{i,l}$ is an independent random variable taking value $\pm\omega_0$ with equal probability. Of course, if it is determined that no price exploration is necessary for product i, then we simply set $p_{i,l} = \hat{p}_{i,l}$.

As a result, in any period, the selling price of a product can be either subject to random price shock, or no price shock. To facilitate our subsequent discussion, at the beginning of each cycle l, we define two sets of cycles for each product $i \in \mathcal{N}$:

$$\begin{aligned} \mathcal{T}_i(l) :=& \{\tau < l \ : i \in S_\tau\}, & t_i(l) :=& |\mathcal{T}_i(l)|, \\ \tilde{\mathcal{T}}_i(l) :=& \{\tau \in \mathcal{T}_i(l) \ : \text{we set } p_{i,l} = p'_{i,l} + \omega_{i,l}\}, & \tilde{t}_i(l) :=& |\tilde{\mathcal{T}}_i(l)|. \end{aligned}$$

In words, $\mathcal{T}_i(l)$ is the set of cycles before cycle l that i is included in the assortment, while $\tilde{\mathcal{T}}_i(l) \subset \mathcal{T}_i(l)$ is the subset of cycles that we impose random price shocks on product i.

The structure of the algorithm. Following our discussion above, the time horizon has been divided into cycles. Next, we further classify the cycles into three categories: i) *insufficiently tested cycle*, ii) *sufficiently tested but insufficiently priced cycle*; and iii) *sufficiently tested and sufficiently priced cycle*. For brevity, we shall simply call the second and third types of cycles *insufficiently priced* and *sufficiently priced* cycles. We refer to Figure 4.1 for a graphical illustration of the three categories of cycles.

The precise definitions of the three categories of cycles are as follows. We first define a product $i \in \mathcal{N}$ as an *insufficiently tested product* at cycle l if it satisfies

$$\tilde{t}_i(l) < c_1 \log(8lN),$$

where c_1 is some constant to be defined later. Then, a cycle l is called an insufficiently

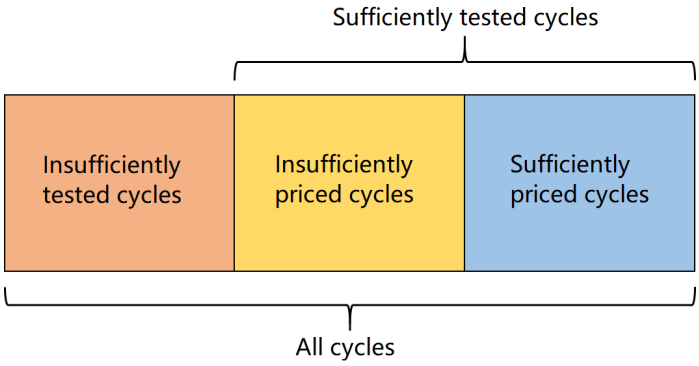


Figure 4.1: Three categories of cycles

tested cycle if there exists at least one product $i \in \mathcal{N}$ that is insufficiently tested at the beginning of cycle l. If all products in cycle l have been sufficiently tested, then l is called a sufficiently tested cycle.

We then classify the products as *insufficiently priced products* and *sufficiently priced products*. Specifically, we call a product i in cycle l an *insufficiently priced product* if it satisfies

$$\lambda_{\min}(W_{i,l}) < c_2 \sqrt{t_i(l) \log(8lN)},$$

where c_2 is some constant to be specified later, and $\lambda_{\min}(W_{i,l})$ is the minimum eigenvalue of the 2×2 symmetric matrix $W_{i,l}$, given by

$$W_{i,l} := \sum_{\tau \in \mathcal{T}_i(l)} \underline{z}_{i,\tau} \underline{z}'_{i,\tau} \in \mathbb{R}^{2\times 2},$$

with

$$\underline{z}'_{i,\tau} := \left(\frac{1}{1 + \exp(\overline{\alpha}_i - \underline{\beta}_i p_{i,\tau})}, -\frac{p_{i,\tau}}{1 + \exp(\overline{\alpha}_i - \underline{\beta}_i p_{i,\tau})} \right). \tag{4.7}$$

Then, a cycle l is called an *insufficiently priced cycle* if all products are sufficiently tested but some product $i \in \mathcal{N}$ is insufficiently priced; if all products are sufficiently tested and also sufficiently priced, then cycle l is called a *sufficiently priced cycle*.

Sufficiently/insufficiently tested/priced cycles are defined as criteria for checking whether more exploration of assortments and prices is needed. The structure of our algorithm is outlined as follows. For an insufficiently tested cycle l, we include in the assortment as many insufficiently tested products as possible and impose a random price shock on all the included products. For an insufficiently priced cycle l, we select as many insufficiently priced products as possible in the assortment and impose random price shock on the insufficiently priced products. Otherwise, i.e., the cycle is sufficiently priced, we select an

assortment and its prices based on a sampled parameter $\tilde{\theta}_l$. After each cycle, i.e., at the end of each iteration in the algorithm, we utilize the collected data to update the posterior parameter distribution, and then repeat this procedure. See Figure 4.2 for a flowchart of our algorithm, and we shall refer to it as *Thompson Sampling with Price Shock (TS-PS)*. The detailed procedure is given in Algorithm 6.

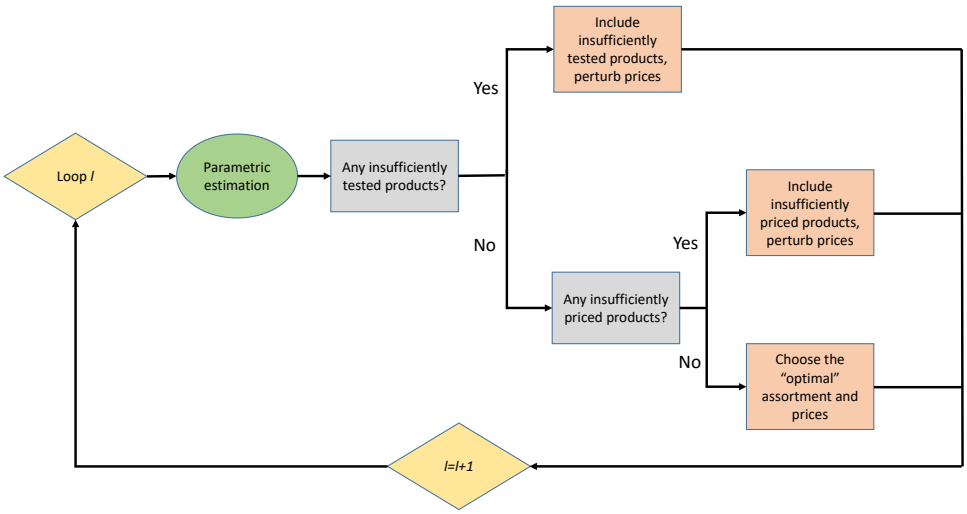


Figure 4.2: The flowchart of algorithm TS-PS

Next we give an overview of the TS-PS algorithm together with an explanation on the intuitions behind each step.

Overview of TS-PS algorithm. As discussed in the introduction section, it is crucial for a learning algorithm to balance the exploration-exploitation trade-off. That is, we need to spend sufficient but not too much time learning the unknown parameter θ, and concurrently exploit the learned information to extract as much revenue as possible. This is achieved through the criterion for sufficiently/insufficiently tested cycles. In addition, to accurately estimate the two choice parameters for each product, the testing prices for each product need to be varied, and that is achieved through the criterion for sufficiently/insufficiently priced cycles. The classification of products and cycles into different categories is done in Step 2 in the TS-PS algorithm. We will show that the total number of insufficiently tested cycles for the algorithm is in the order of $\log T$, while the number of insufficiently priced cycles is in the order of $\sqrt{T}$. In other words, all but $O(\sqrt{T})$ cycles are sufficiently priced cycles that are spent on exploitation.

Algorithm 6 The TS-PS Algorithm

Require: Parameters c_1, c_2; price shock parameter ω_0.

1: **Step 0. Initialization.** Initialize $W_{i,1}$ to be a 2×2 zero matrix and $\mathcal{T}_i(1) = \tilde{\mathcal{T}}_i(1) = \emptyset$ for each product $i \in \mathcal{N}$. Go to Step 1 with $l = 1$.

2: **Step 1. Sampling.** We sample the parameter $\tilde{\theta}_l$ from the posterior distribution Φ_l defined in (4.5). Go to Step 2.

3: **Step 2. Classification of cycle.**

Initialize the set of insufficiently tested products $\mathcal{N}_0(l) = \emptyset$, and initialize the set of insufficiently priced products $\tilde{\mathcal{N}}_0(l) = \emptyset$.

Substep 2.1. Find Insufficiently Tested Products. For each $i \in \mathcal{N}$, if $\tilde{t}_i(l) < c_1 \log(8lN)$, include i in $\mathcal{N}_0(l)$. Go to Substep 2.2.

Substep 2.2. Find Insufficiently Priced Products. For each $i \in \mathcal{N}$, if $\lambda_{\min}(W_{i,l}) < c_2\sqrt{t_i(l)\log(8lN)}$, include i in $\tilde{\mathcal{N}}_0(l)$.

Denote the sufficiently priced products as $\mathcal{N}_1(l) = \mathcal{N} \backslash (\mathcal{N}_0(l) \cup \tilde{\mathcal{N}}_0(l))$. Go to Step 3.

4: **Step 3. Determine Assortment and Price.** From Step 2, we have three cases.

Case 1. If $\mathcal{N}_0(l) \neq \emptyset$, then select $\min\{K, |\mathcal{N}_0(l)|\}$ products uniformly random from $\mathcal{N}_0(l)$ into S_l. Solve for optimal $\hat{p}_l$ given S_l and $\tilde{\theta}_l$ by $\hat{p}_l \in \arg\max_{p\in\mathcal{P}} r(S_l, p, \tilde{\theta}_l)$.

Case 2. If $\mathcal{N}_0(l) = \emptyset$ and $\tilde{\mathcal{N}}_0(l) \neq \emptyset$, then select $\min\{K, |\tilde{\mathcal{N}}_0(l)|\}$ products uniformly random from $\tilde{\mathcal{N}}_0(l)$ into S_l. Solve for optimal $\hat{p}_l$ given S_l and $\tilde{\theta}_l$ by $\hat{p}_l \in \arg\max_{p\in\mathcal{P}} r(S_l, p, \tilde{\theta}_l)$.

Case 3. If $\mathcal{N}_0(l) = \emptyset$ and $\tilde{\mathcal{N}}_0(l) = \emptyset$, then let $(S_l, \hat{p}_l) \in \arg\max_{S\in\mathcal{S}, p\in\mathcal{P}} r(S, p, \tilde{\theta}_l)$.

Note that in all the maximization problems above, if there are multiple optimal solutions, then we select one arbitrarily.

Go to Step 4.

5: **Step 4. Price Shock.** We still consider the three cases in Step 3.

Case 1 and 2. For all $i \in S_l$, define $p'_{i,l} = \text{Proj}_{[\underline{p}_i+\omega_0, \bar{p}_i-\omega_0]}(\hat{p}_{i,l})$. Let $\omega_{i,l} = \pm\omega_0$ a random variable which takes positive and negative value with equal probability. Then set final price to $p_{i,l} = p'_{i,l} + \omega_{i,l}$.

Case 3. For all $i \in S_l$, set $p_{i,l} = \hat{p}_{i,l}$.

Go to Step 5.

6: **Step 5. Implement Decisions.** Initialize $\hat{v}_{i,l} = 0$ for all $i \in \mathcal{N}$. Offer assortment S_l and p_l to customers until the first no-purchase occurs. That is, let E_l denote the set of time periods of cycle l, do

For $t \in E_l$:

Offer (S_t, p_t) to customer t where $S_t = S_l$ and $p_t = p_l$;

Observe customer's choice i_t, and update $\hat{v}_{i_t,l} = \hat{v}_{i_t,l} + 1$;

If $t = T$: exit the algorithm;

End For

For all $i \in S_l$, let $\mathcal{T}_i(l+1) = \mathcal{T}_i(l) \cup \{l\}$. If this cycle belongs to Case 1 or 2, for all $i \in S_l$, let $\tilde{\mathcal{T}}_i(l+l) = \tilde{\mathcal{T}}_i(l) \cup \{l\}$. Let $W_{i,l+1} = W_{i,l} + \underline{z}_{i,l}\underline{z}'_{i,l}$ for all $i \in S_l$, where $\underline{z}_{i,l}$ is given by (4.7).

Update $l = l + 1$ and return to Step 1.

In Steps 3 and 4, we select the assortment and prices to offer to the customer based on the classification of cycle l, and this will guarantee that all products eventually get explored sufficiently and adequately for optimized decisions. Specifically, if cycle l is insufficiently tested or priced, we include those products for further exploration in the assortment, and for insufficiently priced products, we impose price shocks. Although in TS-PS algorithm, we use parameter sampling instead of solving an optimization problem to determine the choice parameters, we will see later that its performance depends critically on and connects closely with the maximum likelihood estimator for each θ_i.

At the end of each cycle, we utilize the observed data and update the posterior distribution of parameter θ, which will be used to sample the parameter $\tilde{\theta}_l$ in (Step 1 of) the next cycle. The main idea behind our use of parameter sampling is the following. The essence of Thompson Sampling is that it replaces the optimization-based estimation (e.g., maximum likelihood estimation) of parameter θ in each iteration l by selecting a $\tilde{\theta}_l$ according to Φ_l. When l is small, it is conceivable that the distribution of Φ_l is not very concentrated around the true parameter θ. Therefore, it is likely that the sampled $\tilde{\theta}_l$ is not close to θ, resulting in suboptimal (S_l, p_l). When l grows large, with the gaining of information the distribution Φ_l will be concentrated around the true value of θ, and the corresponding choice of (S_l, p_l) will be close to the true optimal solution, leading to near-optimal performance.

Remark 4.3.1 *For the TS-PS algorithm, a major computational task is to sample $\tilde{\theta}_l$ according to the posterior distribution Φ_l defined in (4.5) with the joint likelihood function $\mathcal{L}(\theta|\mathcal{F}_{l-1})$ (e.g., the one given in Example 1). Although in our model the posterior distribution Φ_l may not be conjugate to the prior Φ_1, the value of $\tilde{\theta}_l$ can still be sampled quite efficiently with high accuracy using some sampling techniques such as the Metropolis-Hastings algorithm. Briefly speaking, the Metropolis-Hastings algorithm can draw samples from any distribution $P(x)$, provided a function $f(x)$ proportional to $P(x)$ can be calculated. Refer to, e.g., Andrieu et al. (2003). In our numerical experiments in Section 4.4, we will use Metropolis-Hastings algorithm to sample $\tilde{\theta}_l$.*

Remark 4.3.2 *We point out that our TS-PS algorithm is based on Thompson Sampling, but it is not a pure Thompson Sampling algorithm. In our TS-PS algorithm, there are some cycles (i.e., insufficiently tested/priced cycles) that are forced to test some products, while the pure Thompson Sampling algorithm does not have such steps. Therefore, TS-PS can be considered as a modified Thompson Sampling algorithm. These forced exploration cycles are imposed for the purpose of proving theoretical performance. In Section 4.4 we will numerically test the performance of the algorithm when the forced exploration cycles are*

dropped. We note that Cheung and Simchi-Levi (2017) use a pure Thompson Sampling algorithm for personalized assortment optimization. A main difference between our work and theirs is that we adopt a cycle approach that separately estimates the parameters for each product, while Cheung and Simchi-Levi (2017) estimate all parameters together. Hence in terms of the number of product N, our algorithm has Bayesian regret $\tilde{O}(\sqrt{N})$ while Cheung and Simchi-Levi (2017) have Bayesian regret $\tilde{O}(N)$.

4.3.2 Theoretical result

The following result presents the theoretical performance of TS-PS algorithm in terms of Bayesian regret.

Theorem 4.3.1 *Let*

$$c_1 \geq \max\left\{\frac{\max\{1, 1/\overline{p}^4\}4\underline{\kappa}^4\Lambda^2}{\mu^2}, \frac{8\overline{\kappa}^2\underline{\kappa}^2L^2}{\omega_0^2}\right\}, \qquad c_2 \geq \frac{4L}{\mu\Lambda},$$

where $\underline{\kappa}, \overline{\kappa}, \mu, \Lambda, L$ are constants to be specified later in (4.16-4.18) of Section 4.5. Then, there exists some constant c_0 such that the Bayesian regret of TS-PS algorithm is upper bounded by

$$BR(T) \leq c_0\big((1+c_1)N\log(NT) + (1+c_2)\sqrt{NT}\log(NT)\big). \tag{4.8}$$

The proof of Theorem 4.3.1 will be given in Section 5. Briefly, to bound the Bayesian regret of the algorithm we shall separately analyze each of the three categories of cycles: insufficiently tested cycles, insufficiently priced cycles, and sufficiently priced cycles, and derive the Bayesian regret for each of them. In particular, we will prove that the Bayesian regret from insufficiently tested cycles is at most $O(N\log(NT))$, the Bayesian regret for insufficiently priced cycles is at most $O(\sqrt{NT\log(TN)}+N+K\log T)$, and the Bayesian regret for sufficiently priced cycles is at most $O(\sqrt{NT}\log(NT)+K\log T)$.

To gain some intuition on the design of TS-PS algorithm and the performance bound in Theorem 4.3.1, it is important to understand why the cycle approach is adopted, especially given that the TS-PS algorithm (in particular, the sampling of $\tilde{\theta}_l$) does not need to rely on the cycle structure. According to Russo and Van Roy (2014), a crucial step in bounding the Bayesian regret of a Thompson Sampling algorithm is to use its connection with the estimation error of maximum likelihood estimators. Translating to our setting, it means that for the sampled parameter $\tilde{\theta}_l$, the upper bound of the Bayesian regret $\mathbb{E}[r(S^*(\theta), p^*(\theta), \theta) - r(S^*(\tilde{\theta}_l), p^*(\tilde{\theta}_l), \theta)]$ depends on the estimation error of the maximum likelihood estimator $\hat{\theta}_{i,l}$ for each $i \in \mathcal{N}$. Without using a cycle approach, we can only

estimate $\hat{\theta}'_l = (\hat{\theta}'_{1,l}, \ldots, \hat{\theta}'_{N,l})$ jointly using all historical data, and by standard statistical estimation theory (see e.g., Cheung and Simchi-Levi 2017), we obtain an upper bound for the estimation error of $\hat{\theta}_l$, which is proportional to the square root of its dimension $\sqrt{2N}$. This leads to a Bayesian regret $\tilde{O}(N\sqrt{T})$, which is unfavorable when N is large. Using the cycle approach, the maximum likelihood estimations of choice parameters can be decoupled. That is, for each $i \in \mathcal{N}$, the estimator $\hat{\theta}_{i,l}$ depends only on the historical sales data $\hat{v}_{i,\tau}$ of product i, which is the number of purchases of product i in cycle τ for $\tau \in \mathcal{T}_i(l)$. This result attributes to Agrawal et al. (2017a) and is stated below (which holds without their Assumption 4.1).

Lemma 4.3.1 *Conditioned on (S_l, p_l), the moment generating function of $\hat{v}_{i,l}(p_{i,l})$ is given by*

$$\mathbb{E}[e^{\lambda \hat{v}_{i,l}(p_{i,l})} | S_l, p_l] = \frac{1}{1 - v_i(p_{i,l})(e^{\lambda} - 1)}, \quad \lambda \leq \log\left(\frac{1 + v_i(p_{i,l})}{v_i(p_{i,l})}\right), \quad i \in \mathcal{N},$$

where in our setting $v_i(p_{i,l}) = \exp(\alpha_i - \beta_i p_{i,l})$.

This result implies that conditioned on (S_l, p_l), $\hat{v}_{i,l} + 1$ is geometrically distributed with parameter $1/(1 + v_{i,l})$, thus $\mathbb{E}[\hat{v}_{i,l} | S_l, p_l] = v_{i,l}$ and $\hat{v}_{i,l}$ is an unbiased estimator of $v_{i,l}$. Since the marginal distribution of $\hat{v}_{i,l}$ does not depend on the parameters of other products, we can decompose the estimation of choice parameters, and it gives an estimation error proportional to $\sqrt{\log N}$. This allows us to obtain a Bayesian regret upper bound $\tilde{O}(\sqrt{NT})$ and it is the main reason for our adopting the cycle approach. However, it is important to note that, although the marginal distribution of $\hat{v}_{i,l}$ depends only on choice parameters of product i, it does not mean that $\hat{v}_{i,l}$, $i \in S_l$, are independent across products. Therefore, the sampling of parameters $\tilde{\theta}_l$ in cycle l conditioned on history $\mathcal{F}_{l-1}$ has to be done jointly (i.e., using their joint distribution). This explains why we cannot separately sample each $\tilde{\theta}_{i,l}$ using the likelihood function of data $\{\hat{v}_{i,\tau} : \tau \in \mathcal{T}_i(l)\}$ for each product $i \in \mathcal{N}$.

We next elaborate on the definitions for sufficiently/insufficiently tested/priced cycles and their connections with the theoretical result on Bayesian regret. As mentioned earlier, the evaluation of the sampled parameter $\tilde{\theta}_l$ in TS-PS algorithm is through its connection with the maximum likelihood estimator $\hat{\theta}_{i,l}$. Hence it is essential to evaluate the distribution of $\hat{\theta}_{i,l} - \theta_i$ and construct a confidence bound for θ_i, where θ_i is the true choice parameter of product i. The right metric for measuring estimation error turns out to be $||\hat{\theta}_{i,l} - \theta_i||_{\bar{V}_{i,l}}$, where for vector x and positive semidefinite matrix A, $||x||_A := \sqrt{x'Ax}$, and $\bar{V}_{i,l}$ is the

2×2 empirical Fisher's information matrix of product i at cycle l, defined by

$$\bar{V}_{i,l} := I + \sum_{\tau \in \mathcal{T}_i(l)} z_{i,\tau} z'_{i,\tau} \in \mathbb{R}^{2\times 2}, \tag{4.9}$$

where

$$z'_{i,\tau} := \left(\frac{1}{1+\exp(\alpha_i - \beta_i p_{i,\tau})}, -\frac{p_{i,\tau}}{1+\exp(\alpha_i - \beta_i p_{i,\tau})} \right).$$

It will be seen that a key step in constructing a upper bound for $||\hat{\theta}_{i,l} - \theta_i||_{\bar{V}_{i,l}}$ is to develop a concentration inequality for $||Z_{i,l}||_{\bar{V}_{i,l}^{-1}}$, where

$$Z_{i,l} := \sum_{\tau \in \mathcal{T}_i(l)} \epsilon_{i,\tau} z_{i,\tau}, \tag{4.10}$$

and $\epsilon_{i,\tau}$ are random errors of the estimated utilities given by

$$\epsilon_{i,\tau} := \hat{v}_{i,\tau} - v_i, \quad i = 1, 2, \ldots, N, \quad \tau \in \mathcal{T}_i(l). \tag{4.11}$$

Conditioned on (S_l, p_l), the error terms $\epsilon_{i,l}$ are (centered) geometrically distributed, and they are sub-exponential, but not sub-Gaussian, random variables. Thus, to bound the estimation error we need to first establish a concentration inequality for sub-exponential random variables, which is given in Section 5. Note that the results in Agrawal et al. (2017a) cannot be applied in our setting since, our model involves context information for customers, hence a more general concentration inequality with changing context is needed. This new concentration inequality allows us to relate estimation error with sample size. This explains the logic for defining a sufficiently tested product i by $\tilde{t}_i(l) \geq c_1 \log(8lN)$. A similar reasoning holds for defining sufficiently priced product according to $\lambda_{\min}(W_{i,l}) \geq c_2\sqrt{t_i(l)\log(8lN)}$, which follows from the connection between the parameter estimation error and the minimum eigenvalue of $W_{i,l}$. We refer the reader to Section 4.5 for details.

Theorem 4.3.1 presents the performance of TS-PS, which is a Thompson Sampling based algorithm, in terms of Bayesian regret. One question is whether we can characterize its performance in terms of regret, instead of Bayesian regret. Actually, there is a strong connection between the Thompson Sampling algorithm and the Upper-Confidence-Bound (UCB) algorithm (see also Russo and Van Roy 2014). In principle, every Thompson Sampling algorithm can be modified into a UCB algorithm with the same performance in terms of regret. In our setting, the main difficulty comes from the complication in solving the resulting optimization problem. In our TS-PS algorithm, the optimization problem to solve is $\arg\max_{S\in\mathcal{S}, p\in\mathcal{P}} r(S, p, \tilde{\theta}_l)$ which can be computed efficiently (see e.g., Wang 2012). If

we transform it into a UCB algorithm, then let $\hat{\theta}_l = ((\hat{\alpha}_{i,l}, \hat{\beta}_{i,l}) : i \in \mathcal{N})$ be the estimated parameter in cycle l. As a result, in a sufficiently priced cycle l, we will need to solve the following optimization problem:

$$(S_l, p_l) \in \underset{S \subset \mathcal{S}, p \subset \mathcal{P}}{\arg\max} \frac{\sum_{i \in S} w_{i,l}(p_i) p_i}{1 + \sum_{i \in S} w_{i,l}(p_i)}, \tag{4.12}$$

where

$$w_{i,l}(p_i) := e^{\hat{\alpha}_{i,l} - \hat{\beta}_{i,l} p_i} + c_i(l) ||(1, -p_i)||_{V_{i,l}^{-1}}$$

is the estimated utility with upper-confidence bound for some $c_i(l) = O(\sqrt{\log(lN)})$. The term $c_i(l) ||(1, -p_i)||_{V_{i,l}^{-1}}$ is known as the UCB term because with high probability, we have

$$|\exp(\alpha_{i,l} - \beta_{i,l} p_i) - \exp(\hat{\alpha}_{i,l} - \hat{\beta}_{i,l} p_i)| \leq c_i(l) ||(1, -p_i)||_{V_{i,l}^{-1}},$$

hence $\exp(\alpha_{i,l} - \beta_{i,l} p_i) \leq w_{i,l}(p_i)$. If there is an oracle that can be called upon to solve problem (4.12), then we can show that the *regret rate* of this modified UCB algorithm is given by the upper bound in (4.8). Unfortunately, the joint optimization problem (4.12) is very complex because of the term $c_i(l) ||(1, -p_i)||_{\bar{V}_{i,l}^{-1}}$, which makes the optimization problem very hard to solve.

Remark 4.3.3 *Theorem 4.3.1 provides an upper bound for the Bayesian regret of TS-PS algorithm. An immediate question is what is the lower bound for the Bayesian regret of learning algorithms of the joint assortment and pricing optimization problem. We do not have the lower bound for the Bayesian regret but point out that Chen and Wang (2017) study a pure assortment optimization problem, and they obtain an instance-independent lower bound of $\Omega(\sqrt{NT})$ for the minimax regret (not Bayesian regret). Since the pure assortment optimization problem in Chen and Wang (2017) can be considered as a special case of ours by setting $\underline{p}_i = \overline{p}_i = p_i$, our TS-PS algorithm solves their problem with the same Bayesian regret (compared with the regret $O(\sqrt{NT} \log T + N \log^3 T)$ in Agrawal et al. 2017a and $O(\sqrt{NT} \log(KT) + N \log^2(KT))$ in Agrawal et al. 2017b) by simply dropping the insufficiently priced cycles and random price shocks. Thus $\Omega(\sqrt{NT})$ is a lower bound for the minimax regret of our problem as well.*

Remark 4.3.4 *Our method of price exploration via random price shock is similar to the so-called semi-myopic pricing policy (see e.g., Keskin and Zeevi 2014, den Boer 2014) in dynamic pricing problem. In dynamic pricing problem, a sufficient condition for developing an efficient algorithm is to ensure that the empirical Fisher's information matrix has*

sufficiently large minimum eigenvalue. In our problem, we require the minimum eigenvalue of matrix $\bar{V}_{i,l}$ defined in (4.9) to be at least $c_2\sqrt{t_i(l)\log(8lN)}$.

4.4 Numerical Experiments

In this section, we conduct a numerical study on the performance of TS-PS algorithm. For consistency with the literature (e.g., Kaufmann et al. 2012, Bubeck and Liu 2013, Russo and Van Roy 2014), we numerically evaluate the algorithm using regret, even though the theoretical performance of TS-PS is given in terms of Bayesian regret. Another reason for using regret is that in real application, the unknown parameter θ is prefixed, hence the performance using the exact parameter value is more informative than the average performance over its prior or belief distribution. Moreover, we compute the percentage of revenue loss (of an algorithm π), which is the ratio of the regret and the optimal revenue given by

$$L_\theta^\pi(T) := \frac{R_\theta^\pi(T)}{J_\theta^*(T)}.$$

We will demonstrate the effectiveness of the TS-PS algorithm using the percentage of revenue loss compared with the clairvoyant optimal policy.

To the best of our knowledge, this chapter presents the first algorithm for the dynamic joint assortment and pricing problem based on the MNL choice model with unknown parameters. Thus, we do not have algorithms from the literature to directly compare with. Instead, we test several benchmark algorithms, which might be considered by practitioners, described below.

- *Cycle-based Maximum Likelihood Estimation with Price Shock (CMLE-PS)*: This algorithm is exactly the same as TS-PS except that the sampled parameter $\tilde{\theta}_{i,l}$ in each cycle l is replaced by maximum likelihood estimators $\hat{\theta}_{i,l}$ using data $\{\hat{v}_{i,\tau} : \tau \in \mathcal{T}_i(l)\}$, for all $i \in \mathcal{N}$.

- *Epsilon-First:* This is a simple heuristic that first spends certain number of cycles for pure exploration, then spend the rest of the time for exploitation. In each cycle l of the exploration phase, we test an assortment S_l with product prices p_l randomly drawn from their feasible region. After that, the parameters $\hat{\theta}_{i,l}$ are estimated using maximum likelihood method. In the remaining time periods, the "best" assortment and prices are selected based on $\hat{\theta}_i$ for all $i \in \mathcal{N}$ and implemented until the end of the horizon.

- *Epsilon-First with Price Optimization (epsilon-First-PO):* This algorithm can be considered as the combination of epsilon-First and CMLE-PS. Specifically, after pure exploration as in epsilon-First, we fix the "optimal" assortment S_l. In the remaining time periods, we solve a multi-product pricing problem (with fixed S_l) using the random price shock method as in CMLE-PS.

The intuitions for using these three benchmark algorithms are as follows. CMLE-PS is a natural alternative of our TS-PS algorithm because we expect that, when more data is gained, the estimated parameter $\hat{\theta}_l$ will be closer to the real parameter θ so that the revenue will also be closer to the optimal revenue. The second benchmark, epsilon-First, is a classical method used in many online learning problems. This heuristic, though very simple, can sometimes be effective in solving certain problems requiring exploration-exploitation trade-off (see e.g., Tran-Thanh et al. 2010). Finally, in the epsilon-First-PO method, when the assortment is fixed, the problem is reduced to a multi-product pricing problem which has been well-studied.

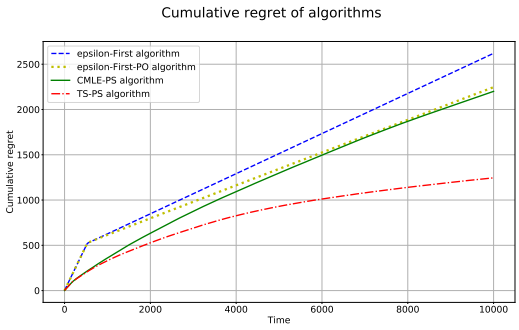
In the following, we first test all the algorithms on a problem instance similar to the one used in Agrawal et al. (2017a). Then, the performance of these algorithms are compared on several randomly generated problem instances. As will be seen, the numerical results show that TS-PS algorithm constantly outperforms the three benchmark algorithms in all the instances we tested, and moreover, its performance is also quite robust across different instances and different tuning parameters.

Problem instance from Agrawal et al. (2017a). We first generate the synthetic data following the numerical experiments in Agrawal et al. (2017a). We consider $N = 10$ products and let the assortment capacity be $K = 4$. For $i \in \mathcal{N}$ such that $i \in \{1, 2, 3, 4\}$, we let $\alpha_i = -1 + \epsilon$ where $\epsilon \in \{0.5, 2.5\}$; for others, we let $\alpha_i = -1$. All products $i \in \mathcal{N}$ have the same price sensitivity $\beta_i = 0.25$. By this construction, the optimal assortment is obviously $S^* = \{1, 2, 3, 4\}$ since they have higher product utility under the same price (see also Wang 2012). The parameter $\epsilon \in \{0.5, 2.5\}$ is to create the "gap" between the optimal revenue and the revenue of any other assortment $S \neq S^*$. We have two choices of ϵ to illustrate the cases of small gap ($\epsilon = 0.5$) and large gap ($\epsilon = 2.5$). The purpose of creating gaps is to show that the performance of TS-PS algorithm is indeed instance-independent, while the performance of other benchmark algorithms might depend on the gap, hence instance-dependent. For other model parameters, we let the feasible regions of parameters and prices be $[\underline{\alpha}, \overline{\alpha}] = [-2, 2]$, $[\underline{\beta}, \overline{\beta}] = [0.1, 0.5]$, $[\underline{p}, \overline{p}] = [0, 20]$, and the price perturbation parameter $\omega_0 = 1$ (this choice is ad-hoc for illustration, as the value of ω_0 only affects the constant term in the regret; and the optimal tuning of this parameter is not the focus of this chapter). The length of time horizon for all the numerical experiments is

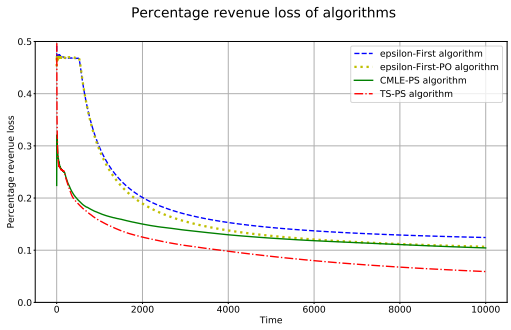
$T = 10,000$, and each algorithm runs 50 times in the experiment and their mean is taken as the final output. For the TS-PS algorithm, a prior distribution needs to be specified and, in each step the posterior distribution has to be computed and sampled. As an illustrative example, we take the prior distribution Φ_1 to be Uniform (thus uninformative) over the feasible region Θ, and use the likelihood function defined in (4.6) for sampling. For the sampling technique, we use the Metropolis-Hastings algorithm (see e.g., Andrieu et al. 2003 for an introduction) with normal (with covariance $0.1I$) as the proposal distribution, and the number of iterations is 2000. The Metropolis-Hastings algorithm has computational complexity linear in time and the sampling iteration, so it gets slower when T becomes large.

For the tuning parameters of our TS-PS algorithm, we remark that the choices of c_1 and c_2 given in Theorem 4.3.1 are for worst-case Bayesian regret bounds. For computational implementation, we do not have to choose them so conservatively. In our numerical experiments, we choose $c_1 = 5$, $c_2 = 0.02$ in TS-PS algorithm (hence in CMLE-PS as well) ad-hoc for both small gap and large gap cases. We note that parameter tuning for learning algorithms is a nontrivial problem and methods have been developed in the literature to tackle it (see e.g., for method of Bayesian optimization, Snoek et al. 2012, Frazier and Wang 2016). Since our focus is not on parameter tuning, we will only present some sensitivity analysis on other combinations of c_1 and c_2 in the TS-PS algorithm at the end of this subsection. For epsilon-First, the only tuning parameter is the number of cycles for exploration. To find a good one, we tested all the numbers from $\{100, 200, 300, \ldots, 2000\}$ for the small gap case, and found that 400 is the best choice. Similarly, we found that 400 periods of pure exploration is also the best choice for epsilon-First-PO (which, after fixing the assortment, applies a multi-product pure pricing algorithm as in CMLE-PS with $c_2 = 0.02$). We then fix input parameters of all algorithms for both small and large gap cases. The reason that we do not separately tune the parameter for two cases is that we want to test the robustness of algorithms under different problem instances (indeed, in the real application we may not have the opportunity for specific parameter tuning for each instance).

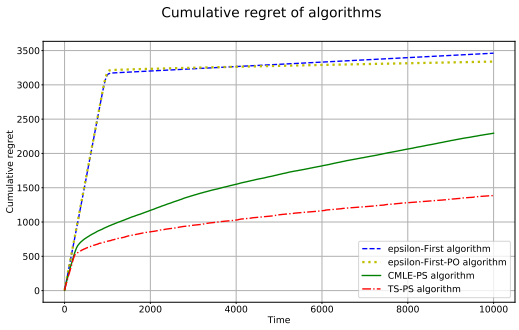
The numerical results of TS-PS and the benchmark algorithms are summarized in Figure 4.3. Note that for both $\epsilon = 0.5$ and $\epsilon = 2.5$, TS-PS (the dash-dotted line) clearly has the best performance. CMLE-PS (the solid line) has good performance in the case of $\epsilon = 2.5$, but performs relatively poorly when $\epsilon = 0.5$. This discrepancy shows that the performance of CMLE-PS is dependent on problem instances. The reason for this discrepancy is that for CMLE-PS, if the number of insufficiently tested cycles is not enough for the small gap, it cannot accurately identify the optimal assortment S^*; hence it repeatedly chooses sub-



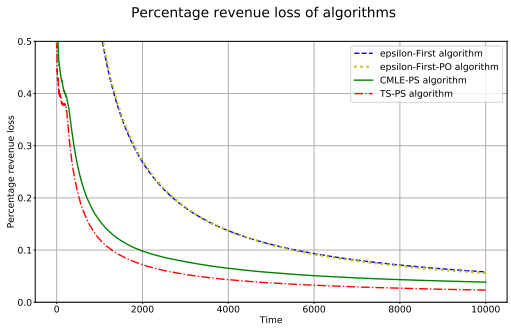
(a) Cumulative regrets with $\epsilon = 0.5$



(b) Percentage revenue loss with $\epsilon = 0.5$



(c) Cumulative regrets with $\epsilon = 2.5$



(d) Percentage revenue loss with $\epsilon = 2.5$

Figure 4.3: Performances of algorithms in small gap and large gap cases

	epsilon-First			epsilon-First-PO		
	10th percentile	Mean	90th percentile	10th percentile	Mean	90th percentile
$\epsilon = 0.5$	4.14%	12.43%	22.39%	4.08%	10.66%	17.14%
$\epsilon = 2.5$	5.21%	5.81%	6.39%	5.23%	5.61%	6.16%

	CMLE-PS			TS-PS		
	10th percentile	Mean	90th percentile	10th percentile	Mean	90th percentile
$\epsilon = 0.5$	4.62%	10.44%	21.00%	4.71%	5.91%	7.40%
$\epsilon = 2.5$	2.00%	3.85%	7.15%	2.09%	2.33%	2.76%

Table 4.1: The mean, 10th percentile, and 90th percentile of percentage of revenue loss for all algorithms when $\epsilon = 0.5$ and $\epsilon = 2.5$.

optimal assortments in other cycles. The other two benchmarks, epsilon-First (the dashed line) and epsilon-First-PO (the dotted line), have quite similar (poor) performances in both cases. As observed from their curves, we can see that when $\epsilon = 0.5$ is small, 400 cycles of pure exploration cannot identify S^* very well, causing the subsequent revenue loss. On the other hand, when $\epsilon = 2.5$, though the algorithm can identify S^* quite well after pure

exploration, the revenue loss during the exploration cycles is very significant (because of the large gap), making the overall performance quite bad. Moreover, comparing epsilon-First with epsilon-First-PO, we observe that price optimization after fixing assortment does bring some benefit, but this benefit is not very significant compared with the revenue loss.

In Table 4.1, we present the results on the total revenue for the mean, 10th percentile, and 90th percentile of samples of the percentage of revenue loss for each algorithm. From these results, it can be seen that, compared with other benchmarks, TS-PS is much more robust. For instance, when $\epsilon = 0.5$, the difference between 10th and 90th percentile of TS-PS is only 2.69%, while it is more than 13% for other algorithms.

	$c_2 = 0.02$			$c_2 = 0.06$		
	10th percentile	Mean	90th percentile	10th percentile	Mean	90th percentile
$c_1 = 5$	4.71%	5.91%	7.40%	5.19%	6.19%	7.56%
$c_1 = 30$	6.34%	7.25%	8.37%	6.54%	7.20%	8.13%

Table 4.2: The mean, 10th percentile, and 90th percentile of percentage of revenue loss for TS-PS (when $\epsilon = 0.5$) with different combinations of c_1 and c_2.

To understand the sensitivity of TS-PS algorithm with respect to the tuning parameters c_1 and c_2, we tested all combinations of $c_1 \in \{5, 30\}$ and $c_2 \in \{0.02, 0.06\}$ for $\epsilon = 0.5$ (since the algorithm performance with $\epsilon = 0.5$ has more variation than that with $\epsilon = 2.5$ as seen from Table 4.1), and the results are summarized in Table 4.2. We observe that performance of TS-PS algorithm is quite consistent (and all better than other methods) in all combinations of c_1 and c_2, though it generally performs somewhat better when c_1 and c_2 are smaller. This shows the robustness of the algorithm on the tuning parameters.

The theoretical performance of TS-PS algorithm in Theorem 4.3.1 is obtained based on lower bound conditions on tuning parameters c_1 and c_2. One question that arises is, how does the algorithm perform when these conditions are not satisfied, e.g., $c_1 = c_2 = 0$ (i.e., pure Thompson Sampling)? We did another numerical experiment with $\epsilon = 0.5$ and $c_1 = c_2 = 0$, and found that the mean percentage revenue loss is 6.27% with 10th percentile 4.97% and 90th percentile 7.31%. This shows that the TS-PS algorithm still performs very well, though slightly worse than that with $c_1 = 5, c_2 = 0.02$ or $c_1 = 5, c_2 = 0.06$. We draw two insights from this test. First, appropriate forced exploration may still be preferred to obtain the best empirical performance, and second, pure Thompson Sampling algorithm also works well for joint pricing and assortment optimization. It is not clear to us whether the theoretical performance of the algorithm remains to hold when the conditions on c_1 and c_2 are dropped.

Randomly generated instances. In this section, we present the numerical results of

TS-PS and the three benchmark algorithms on four randomly generated instances. These problems have the same feasible regions for parameters Θ and prices $\mathcal{P}$ as in the previous instances, and the parameters θ of the four instances are generated uniformly from Θ. For brevity, we will not present the regret curves and only summarize the numerical results on the percentage of revenue loss in Table 4.3.

	epsilon-First			epsilon-First-PO		
	10th percentile	Mean	90th percentile	10th percentile	Mean	90th percentile
Instance 1	8.39%	9.20%	10.31%	7.88%	8.80%	9.85%
Instance 2	4.63%	5.94%	8.34%	4.41%	5.00%	5.86%
Instance 3	6.18%	7.25%	9.08%	5.92%	6.61%	8.23%
Instance 4	5.39%	6.42%	7.46%	5.27%	5.98%	6.96%

	CMLE-PS			TS-PS		
	10th percentile	Mean	90th percentile	10th percentile	Mean	90th percentile
Instance 1	2.44%	3.90%	5.84%	2.01%	2.24%	2.50%
Instance 2	2.47%	4.56%	8.13%	2.44%	2.80%	3.50%
Instance 3	2.61%	6.25%	13.10%	2.09%	2.32%	2.67%
Instance 4	2.26%	4.09%	6.35%	2.43%	2.85%	3.32%

Table 4.3: The mean, 10th percentile, and 90th percentile of percentage of revenue loss for all algorithms on four random instances.

We observe similar results as in the previous testings. The TS-PS algorithm has the best performance for all instances, and its performances have very small variations as seen from its 10th and 90th percentiles. So once again, the numerical experiments show that TS-PS algorithm has instance-independent performance, and that it consistently outperforms the three benchmark algorithms.

4.5 Proof of Main Result

In this section, we present the main steps in the proof of Theorem 4.3.1. As mentioned in Section 4.3.2, a critical step to analyze the Bayesian regret is to evaluate the confidence bound $||\hat{\theta}_{i,l}-\theta_i||_{\bar{V}_{i,l}}$, where $\hat{\theta}_{i,l}$ is the maximum likelihood estimator and $\bar{V}_{i,l}$ is the empirical Fisher's information matrix of product i in cycle l defined in (4.9). This confidence bound further requires a concentration inequality result for $||Z_{i,l}||_{\bar{V}_{i,l}^{-1}}$, where $Z_{i,l}$ is a sum of sub-exponential random errors defined in (4.10). Therefore, in this section we first present the confidence bound and a general concentration inequality result for sub-exponential random variables in Section 4.5.1, for which $||Z_{i,l}||_{\bar{V}_{i,l}^{-1}}$ is a special case. Then, in Section 4.5.2, we

prove Theorem 4.3.1. All the missing proofs of the technical results can be found in the section of proofs.

4.5.1 Confidence bound and concentration inequalities

The proof of Theorem 4.3.1 is based on the relationship between the sampled parameter $\tilde{\theta}_l$ and the maximum likelihood estimator (see e.g., Russo and Van Roy 2014). By Lemma 4.3.1, the likelihood function of θ_i of product i before cycle l with history $\mathcal{F}_{l-1}$, which includes $\hat{v}_{i,\tau}(p_{i,\tau})$, is

$$\mathcal{L}_i(\phi|\mathcal{F}_{l-1}) := \prod_{\tau \in \mathcal{T}_i(l)} \left(\frac{e^{\phi' x_{i,\tau}}}{1 + e^{\phi' x_{i,\tau}}} \right)^{\hat{v}_{i,\tau}} \frac{1}{1 + e^{\phi' x_{i,\tau}}}.$$

The maximum likelihood estimator $\hat{\theta}_{i,l}$ for product i can be computed separately from other products by

$$\hat{\theta}_{i,l} \in \arg\max_{\phi \in \Theta_i} \mathcal{L}_i(\phi|\mathcal{F}_{l-1}).$$

Note that $\arg\max_{\phi \in \Theta_i} \mathcal{L}_i(\phi|\mathcal{F}_{l-1})$ exists because, by its definition, $\mathcal{L}_i(\phi|\mathcal{F}_{l-1})$ is obviously a continuous function of ϕ, and Θ_i is a compact set in $\mathbb{R}^2$. Then an important upper bound for $||\hat{\theta}_{i,l} - \theta_i||_{\bar{V}_{i,l}}$ is proved in the following proposition.

Proposition 4.5.1 *For any $i \in \mathcal{N}$ and l a sufficiently priced loop, we have*

$$||\hat{\theta}_{i,l} - \theta_i||_{\bar{V}_{i,l}} \leq k_1 \sqrt{\log(lN)} \tag{4.13}$$

for some constant k_1 with probability at least $1 - 1/(lN)$.

The rest of this subsection is to show the main steps of proving the above result (with all the details in the section of proofs). The main technical step to prove Proposition 4.5.1 is an important concentration inequality of sub-exponential random variable. In particular, we want to bound $||Z_{i,l}||_{\bar{V}_{i,l}^{-1}}$.

Recall that a random variable X is said to be sub-exponential if there exists a constant $b > 0$, such that for all $a \geq 0$,

$$\mathbb{P}(|X| \geq a) \leq 2e^{-ba}.$$

It is known (see e.g., Wainwright 2019) that X is sub-exponential if and only if there exist positive parameters (Λ, μ), such that for any λ with $|\lambda| \leq \Lambda$, it satisfies

$$\mathbb{E}[e^{\lambda X}] \leq e^{\mu^2 \lambda^2 / 2}.$$

To bound $||Z_{i,l}||_{\bar{V}_{i,l}^{-1}}$, we need to resolve two main technical difficulties. First, the ordinary concentration inequalities such as Azuma's inequality cannot be applied in our setting because the term $\bar{V}_{i,l}^{-1}$ makes the errors dependent on each other, i.e., the sequence

$$||Z_{i,l}||_{\bar{V}_{i,l}^{-1}} = \sqrt{\Big(\sum_{\tau\in\mathcal{T}_i(l)} \epsilon_{i,\tau} z'_{i,\tau}\Big)\Big(I + \sum_{\tau\in\mathcal{T}_i(l)} z_{i,\tau} z'_{i,\tau}\Big)^{-1}\Big(\sum_{\tau\in\mathcal{T}_i(l)} \epsilon_{i,\tau} z_{i,\tau}\Big)}$$

cannot be written as the sum of some adapted random errors for applying concentration inequalities (see e.g., Auer 2002, for more detailed discussion of this issue). Second, to the best of our knowledge, all the concentration inequalities established in the literature (see e.g., Abbasi-Yadkori et al. 2011, Cheung and Simchi-Levi 2017, Li et al. 2017a) require the random errors to be sub-Gaussian, but in our problem $\epsilon_{i,l}$ are only sub-exponential, not sub-Gaussian. We resolve these challenges using the following proposition.

Proposition 4.5.2 *Let $\{\mathcal{F}_t\}_{t=1}^{\infty}$ be a filtration, and $\{\epsilon_t\}_{t=1}^{\infty}$ be a real-valued stochastic process such that ϵ_t is $\mathcal{F}_t$-measurable and ϵ_t is conditionally sub-exponential with parameters (Λ, μ) for all t, i.e., for any λ satisfying $|\lambda| \leq \Lambda$,*

$$\mathbb{E}[e^{\lambda\epsilon_t}|\mathcal{F}_{t-1}] \leq e^{\mu^2\lambda^2/2}, \quad t = 1, 2, \ldots.$$

Let $\{z_t\}_{t=1}^{\infty}$ be an $\mathbb{R}^d$-valued stochastic process such that z_t is $\mathcal{F}_{t-1}$-measurable and $||z_t||_2^2 \leq L$ for some constant $L > 0$. Let V be an arbitrary $d \times d$ positive definite matrix. For any $t > 0$, define

$$\bar{V}_t = V + \sum_{s=1}^{t-1} z_s z'_s, \qquad Z_t = \sum_{s=1}^{t-1} \epsilon_s z_s. \tag{4.14}$$

Suppose τ is a stopping time with respect to the filtration $\{\mathcal{F}_t\}_{t=1}^{\infty}$, and define the event

$$\mathcal{E}_\tau = \big\{||Z_\tau||_2/\lambda_{\min}(\bar{V}_\tau) \leq \mu^2\Lambda/(2\sqrt{L})\big\}. \tag{4.15}$$

Then we have that for any $\delta > 0$,

$$\mathbb{P}\left(||Z_\tau||^2_{\bar{V}_\tau^{-1}} > 2\mu^2 \log\left(\frac{\det(\bar{V}_\tau)^{1/2}}{\delta k(\mu, \Lambda, L, V)\det(V)^{1/2}}\right), \mathcal{E}_\tau\right) < \delta,$$

where $k(\mu, \Lambda, L, V)$ is a positive constant depending on μ, Λ, L, V.

This proposition is similar in spirit to Theorem 1 in Abbasi-Yadkori et al. (2011), which develops a concentration inequality for sub-Gaussian errors. Compared to that result, a key difference is that the bound for sub-exponential random errors is on event (4.15). The

intuition is that locally, sub-exponential distribution has similar bound on its moment generating function as a sub-Gaussian distribution, which plays an extremely important role in developing concentration inequalities. Specifically, by the definition of sub-Gaussian distribution, for any real λ, it satisfies $\mathbb{E}[\exp(\lambda X)] \leq \exp(\mu^2\lambda^2/2)$. For sub-exponential distribution, however, this inequality is satisfied only on $|\lambda| \leq \Lambda$ for some $\Lambda > 0$. As a result, the concentration inequality can be established only on the event (4.15) (which is nonetheless sufficient for our regret analysis).

An immediate corollary of Proposition 4.5.2, by specifying the bound $\det(\bar{V}_t) \leq (1 + \tau L/d)^d$ with $\bar{V}_t$ defined in (4.14) and setting $V = I$, is the following result (see Lemma 10 of Abbasi-Yadkori et al. 2011 for a similar result for sub-Gaussian errors).

Corollary 4.5.1 *When the random errors satisfy the conditions in Proposition 4.5.2, we have, for any $\delta > 0$,*

$$\mathbb{P}\left(||Z_\tau||_{\bar{V}_\tau^{-1}} > \mu\sqrt{d\log(1+\tau L/d) + d\log\left(1/(\delta k(\mu,\Lambda,L,V))\right)}, \mathcal{E}_\tau\right) < \delta.$$

We now demonstrate how to (conditionally) bound $||Z_{i,l}||_{\bar{V}_{i,l}^{-1}}$ using Proposition 4.5.2 and Corollary 4.5.1. As mentioned earlier, the estimation errors $\epsilon_{i,l}$ defined in (4.11) are sub-exponential because $\hat{v}_{i,l}+1$ are geometrically distributed with mean $v_{i,l}+1$. By Lemma 4.3.1, for given θ_i and price $p_{i,l}$, there exist constants $\mu(\theta_i, p_{i,l})$ and $\Lambda(\theta_i, p_{i,l})$, that depend on θ_i and $p_{i,l}$, such that for any $i \in \mathcal{N}$ and $l < L_T$, it holds that

$$\mathbb{E}[e^{\epsilon_{i,l}\lambda}|\mathcal{F}_{l-1}] \leq e^{\mu(\theta_i,p_{i,l})^2\lambda^2/2}, \qquad |\lambda| \leq \Lambda(\theta_i, p_{i,l}).$$

Since the prices $p_{i,l}$ and parameters θ_i are all bounded, there must exist constants μ and Λ that satisfy

$$\mu \geq \mu(\theta_i, p_{i,l}),\ \Lambda \leq \Lambda(\theta_i, p_{i,l}),\ \text{for all } \theta_i \in \Theta_i, p_{i,l} \in [\underline{p}_i, \overline{p}_i],\ \ i \in \mathcal{N}, l \in \{1,\ldots,L_T\}. \tag{4.16}$$

Therefore, for any $i \in \mathcal{N}$ and $l < L_T$, it holds that

$$\mathbb{E}[e^{\epsilon_{i,l}\lambda}|\mathcal{F}_{l-1}] \leq e^{\mu^2\lambda^2/2}, \qquad \text{for all } |\lambda| \leq \Lambda. \tag{4.17}$$

To present an upper bound for $||z_{i,l}||_2^2$, we define notations

$$L := (1+\overline{p}^2)/\underline{\kappa}^2, \quad \underline{\kappa} := 1 + e^{\underline{\alpha}-\overline{\beta}\overline{p}}, \quad \overline{\kappa} := 1 + e^{\overline{\alpha}-\underline{\beta}\underline{p}}. \tag{4.18}$$

Then $||z_{i,l}||_2^2 \leq L$. Applying Corollary 4.5.1 with $\tau = t_i(l)$ and $\delta = 1/(2lN)$, we obtain

the following result for $||Z_{i,l}||_{\bar{V}_{i,l}^{-1}}$.

Corollary 4.5.2 *For any $l > 1$, define the event*

$$\mathcal{E}_{i,l} = \left\{||Z_{i,l}||_2/\lambda_{\min}(\bar{V}_{i,l}) \leq \mu^2\Lambda/(2\sqrt{L})\right\}. \tag{4.19}$$

Then we have

$$\mathbb{P}\left(||Z_{i,l}||_{\bar{V}_{i,l}^{-1}} > \mu\sqrt{2\log(1+t_i(l)L/2) + 2\log\left(k_2 lN\right)}, \mathcal{E}_{i,l}\right) < 1/(2lN),$$

where $k_2 := 2/k(\mu, \Lambda, L, I)$ and $k(\mu, \Lambda, L, I)$ is given in Proposition 4.5.2.

Thus, Proposition 4.5.1 is proved using Corollary 4.5.2. In particular, we can show that conditioned on $||Z_{i,l}||_{\bar{V}_{i,l}^{-1}} \leq \mu\sqrt{2\log(1+t_i(l)L/2) + 2\log\left(k_2 lN\right)}$, we have $||\hat{\theta}_{i,l} - \theta_i||_{\bar{V}_{i,l}} \leq k_1\sqrt{\log(lN)}$. The detailed argument can be found in the section of proofs.

4.5.2 Proof of Theorem 4.3.1

The cumulative Bayesian regret of the TS-PS algorithm can be written as (recall that L_T is the index of the last cycle)

$$\begin{aligned}
&\mathbb{E}\left[\sum_{l=1}^{L_T}|E_l|(r(S^*,p^*) - r(S_l,p_l))\right] = \mathbb{E}\left[\sum_{l=1}^{L_T}\mathbb{E}[|E_l|(r(S^*,p^*) - r(S_l,p_l))|S_l,p_l]\right] \\
\leq&\mathbb{E}\left[\sum_{l=1}^{L_T}(1+\sum_{i\in S_l}e^{\alpha_i-\beta_i p_{i,l}})(r(S^*,p^*) - r(S_l,p_l))\right] \leq \overline{\kappa}\mathbb{E}\left[\sum_{l=1}^{L_T}|S_l|(r(S^*,p^*) - r(S_l,p_l))\right],
\end{aligned}$$

where $\overline{\kappa}$ is a constant defined in (4.18), and E_l is the set of periods in cycle l, i.e., $E_l := \{t_l, t_l+1, \ldots, t_{l+1}-1\}$ for $l < L_T$ and $E_{L_T} := \{t_{L_T}, t_{L_T}+1, \ldots, T\}$. Note that the first inequality follows that by definition, $|E_l|$ (for all $l < L_T$) is geometrically distributed with parameter $1/(1+\sum_{i\in S_l}e^{\alpha_i-\beta_i p_{i,l}})$. For $l = L_T$, by definition of L_T, in the last period T it might be possible that $i_T \neq 0$, so we have that $E[|E_{L_T}||S_{L_T}, p_{L_T}] \leq 1/(1+\sum_{i\in S_{L_T}}e^{\alpha_i-\beta_i p_{i,l}})$.

To bound the Bayesian regret, we consider the three categories of cycles separately: the set of insufficiently tested cycles (denoted by $\mathcal{C}_0$), the set of insufficiently priced cycles

(denoted by $\tilde{\mathcal{C}}_0$), and the set of sufficiently priced cycles (denoted by $\mathcal{C}_1$). That is,

$$\begin{aligned}
\mathcal{C}_0 :=& \{l \le L_T : \text{for some } i \in \mathcal{N}, \tilde{t}_i(l) < c_1 \log(8lN)\};\\
\tilde{\mathcal{C}}_0 :=& \{l \le L_T : \tilde{t}_i(l) \ge c_1 \log(8lN) \quad \forall i \in \mathcal{N},\\
& \quad \text{but for some } i \in \mathcal{N}, \lambda_{\min}(W_{i,l}) < c_2\sqrt{t_i(l)\log(8lN)}\};\\
\mathcal{C}_0 :=& \{l \le L_T : \tilde{t}_i(l) \ge c_1 \log(8lN), \lambda_{\min}(W_{i,l}) \ge c_2\sqrt{t_i(l)\log(8lN)} \quad \forall i \in \mathcal{N}\}.
\end{aligned}$$

Bayesian regret from insufficiently tested cycles. By definition, for any cycle $l \in \mathcal{C}_0$ and any product $i \in S_l$, it holds that $\tilde{t}_i(l) < c_1 \log(8lN)$. Since all products in $\mathcal{C}_0$ have price shock, we have

$$\sum_{l\in\mathcal{C}_0} |S_l| \le \sum_{i\in\mathcal{N}} (\tilde{t}_i(l_{0,i}) + 1) < c_1 N \log(8TN) + N,$$

where $l_{0,i}$ is the last insufficiently tested cycle in which product i is included in the assortment, i.e., $i \in S_{l_{0,i}}$ for $l_{0,i} \in \mathcal{C}_0$, but for any $l > l_{0,i}$ such that $l \in \mathcal{C}_0$, $i \notin S_l$. To see the above inequality, first note that by the definition of $\mathcal{C}_0$, for any $l \in \mathcal{C}_0$, the selected products in assortment S_l are all insufficiently tested. Since in each $l \in \mathcal{C}_0$, product $i \in S_l$ has price shock, $\sum_{l\in\mathcal{C}_0} |S_l|$ is bounded above by the sum of number of price shocks of each product i until its last time selected in $S_{l_{0,i}}$ with $l_{0,i} \in \mathcal{C}_0$ (including period $l_{0,i}$). This number is exactly $\tilde{t}_i(l_{0,i}) + 1$ (recall that, by definition, $\tilde{t}_i(l_{0,i})$ does not include period $l_{0,i}$).

This shows that, the cumulative Bayesian regret of insufficiently tested cycles is at most $(c_1 + 1)N \log(8TN)$.

Bayesian regret from insufficiently priced cycles. To bound the Bayesian regret of insufficiently priced cycles, we first define an important event

$$\mathcal{E}_{V_{i,l}} := \left\{ \lambda_{\min}(W_{i,l}) \ge \frac{\tilde{t}_i(l)\omega_0^2}{2\overline{\kappa}^2\underline{\kappa}^2 L} \right\},$$

which gives a lower bound of $\lambda_{\min}(W_{i,l})$. This event holds with probability at least $1 - 1/(2lN)$ by the following lemma, which is proved in the section of proofs.

Lemma 4.5.1 *For any product $i \in \mathcal{N}$ in cycle l with $\tilde{t}_i(l) \ge c_1 \log(8lN)$, when $c_1 \ge 8\overline{\kappa}^2\underline{\kappa}^2 L^2/\omega_0^2$, we have*

$$\lambda_{\min}\Big(\sum_{\tau\in\mathcal{T}_i(l)} \underline{z}_{i,\tau}\underline{z}'_{i,\tau} \Big) \ge \frac{\tilde{t}_i(l)\omega_0^2}{2\overline{\kappa}^2\underline{\kappa}^2 L}$$

with probability at least $1 - 1/(2lN)$.

Denote $\tilde{C}_{0,i}(l)$ as the number of insufficiently priced cycles before l in which i is in-

cluded in the assortment, i.e. $\tilde{C}_{0,i}(l) := \left|\{\tau \in \tilde{\mathcal{C}}_0 : \tau < l, i \in S_\tau\}\right|$. Note that we can bound the Bayesian regret of insufficiently tested cycles by

$$\overline{\kappa}\mathbb{E}\left[\sum_{l\in\tilde{\mathcal{C}}_0}|S_l|(r(S^*,p^*)-r(S_l,p_l))\right] \le \overline{\kappa p}\mathbb{E}\left[\sum_{l\in\tilde{\mathcal{C}}_0}|S_l|\right].$$

Define the event $\mathcal{E}_{V_l} := \bigcap_{i\in\mathcal{N}}\mathcal{E}_{V_{i,l}}$, we have

$$\begin{aligned}
\mathbb{E}\left[\sum_{l\in\tilde{\mathcal{C}}_0}|S_l|\right] =& \mathbb{E}\left[\sum_{l\in\tilde{\mathcal{C}}_0}|S_l|\mathbf{1}(\mathcal{E}_{V_l})\right] + \mathbb{E}\left[\sum_{l\in\tilde{\mathcal{C}}_0}|S_l|\mathbf{1}(\mathcal{E}'_{V_l})\right] \\
\le& \mathbb{E}\left[\sum_{i\in\mathcal{N}}(\tilde{C}_{0,i}(l_{1,i})+1)\mathbf{1}(\mathcal{E}_{V_{l_{1,i}}})\right] + K\sum_{l=1}^{L_T}\frac{1}{2l} \\
\le& \mathbb{E}\left[\sum_{i\in\mathcal{N}}\tilde{C}_{0,i}(l_{1,i})\mathbf{1}(\mathcal{E}_{V_{l_{1,i}}})\right] + N + K(\log(T)+1)/2,
\end{aligned}$$

where $l_{1,i}$ denotes the last insufficiently priced cycle in which event $\mathcal{E}_{V_{l_{1,i}}}$ holds and $i \in S_{l_{1,i}}$, and $\mathcal{A}'$ is the complement of event $\mathcal{A}$. The first inequality above is from the union bound of events $\mathcal{E}'_{V_{i,l}}$ for all $i \in \mathcal{N}$ such that $\mathcal{E}'_{V_l}$ holds with probability at most $N/(2lN) = 1/(2l)$ in any insufficiently priced loop l.

To bound $\mathbb{E}\left[\sum_{i\in\mathcal{N}}\tilde{C}_{0,i}(l_{1,i})\mathbf{1}(\mathcal{E}_{V_{l_{1,i}}})\right]$, we note that on $\mathcal{E}_{V_{l_{1,i}}}$, because $\tilde{C}_{0,i}(l_{1,i}) \le \tilde{t}_i(l_{1,i})$ by definition, $\lambda_{\min}(W_{i,l_{1,i}}) \ge \tilde{C}_{0,i}(l_{1,i})\omega_0^2/(2\overline{\kappa}^2\underline{\kappa}^2 L), i \in \mathcal{N}$; since $l_{1,i} \in \tilde{\mathcal{C}}_0$, by definition, $\lambda_{\min}(W_{i,l_{1,i}}) < c_2\sqrt{t_i(l_{1,i})\log(8l_{1,i}N)}$ for all $i \in \mathcal{N}$. As a result, on event $\mathcal{E}_{V_{l_{1,i}}}$, we have

$$\tilde{C}_{0,i}(l_{1,i}) < c_2(2\overline{\kappa}^2\underline{\kappa}^2 L)\sqrt{t_i(l_{1,i})\log(8l_{1,i}N)}/\omega_0^2 \le c_2(2\overline{\kappa}^2\underline{\kappa}^2 L)\sqrt{t_i(L_T)\log(8TN)}/\omega_0^2. \tag{4.20}$$

Combining (4.20) with $\mathbb{E}\left[\sum_{i\in\mathcal{N}}\tilde{C}_{0,i}(l_{1,i})\mathbf{1}(\mathcal{E}_{V_{l_{1,i}}})\right]$, we obtain

$$\begin{aligned}
\mathbb{E}\left[\sum_{i\in\mathcal{N}}\tilde{C}_{0,i}(l_{1,i})\mathbf{1}(\mathcal{E}_{V_{l_{1,i}}})\right] \le& c_2k_4\sqrt{\log(TN)}\sum_{i\in\mathcal{N}}\sqrt{\mathbb{E}[t_i(L_T)]} \\
\le& c_2k_4\sqrt{NT\log(TN)/(\underline{\kappa}-1)}
\end{aligned}$$

for some constant k_4 defined such that $(2\overline{\kappa}^2\underline{\kappa}^2 L)\sqrt{\log(8TN)}/\omega_0^2 \le k_4\sqrt{\log(TN)}$. In the inequalities above, the first inequality is from (4.20) and Jensen's inequality, and the last inequality follows from $\sum_{i\in\mathcal{N}}\mathbb{E}[t_i(L_T)] \le T/(\underline{\kappa}-1)$ (see (A.20) in the proof of Theorem 1 of Agrawal et al. 2017a; note that this result does not depend on Assumption

4.1.1 in Agrawal et al. 2017a, so it is applicable to our setting), and $\sum_{i\in\mathcal{N}}\sqrt{\mathbb{E}[t_i(L_T)]} \leq \sqrt{N\sum_{i\in\mathcal{N}}\mathbb{E}[t_i(L_T)]} \leq \sqrt{NT/(\underline{\kappa}-1)}$. This shows that, the expected regret of insufficiently priced cycles is at most $O(c_2\sqrt{NT\log(TN)}+N+K\log T)$.

Bayesian regret from sufficiently priced cycles. The proof of this part borrows ideas from Russo and Van Roy (2014). Conditioning on $\mathcal{F}_{l-1}$, it follows from the definition of posterior sampling in (4.5) that θ_l has the same distribution as θ (see e.g., Russo and Van Roy 2014). Therefore, the chosen (S_l, p_l) according to θ_l is also identically distributed as (S^*, p^*) (which is chosen according to θ) given $\mathcal{F}_{l-1}$. The next step is from the so-called UCB function $U_l(S,p): \mathcal{S}\times\mathcal{P}\to\mathbb{R}$ which is defined as

$$U_l(S,p) := r(S,p,\hat{\theta}_l) + \frac{2\bar{p}(\overline{\kappa}-1)}{|S|(\underline{\kappa}-1)}\sum_{i\in S}||\hat{\theta}_{i,l}-\theta_i||_{\bar{V}_{i,l}}||(1,-p_i)||_{\bar{V}_{i,l}^{-1}}. \tag{4.21}$$

We have that

$$\begin{aligned}
&\mathbb{E}[r(S^*,p^*,\theta)-r(S_l,p_l,\theta)] = \mathbb{E}[\mathbb{E}[r(S^*,p^*,\theta)-r(S_l,p_l,\theta)|\mathcal{F}_{l-1}]]\\
=&\mathbb{E}[\mathbb{E}[r(S^*,p^*,\theta)-U_l(S^*,p^*)+U_l(S_l,p_l)-r(S_l,p_l,\theta)|\mathcal{F}_{l-1}]]\\
=&\mathbb{E}[r(S^*,p^*,\theta)-U_l(S^*,p^*)]+\mathbb{E}[U_l(S_l,p_l)-r(S_l,p_l,\theta)],
\end{aligned} \tag{4.22}$$

where the second equality is because (S^*,p^*) and (S_l,p_l) are identically distributed given $\mathcal{F}_{l-1}$ and $\mathbb{E}[U_l(S_l,p_l)|\mathcal{F}_{l-1}] = \mathbb{E}[U_l(S^*,p^*)|\mathcal{F}_{l-1}]$ by definition of $U_l(S,p)$ (as $U_l(S,p)$ is a deterministic function of (S,p) given history $\mathcal{F}_{l-1}$).

To proceed, we need the following lemma, which is proved in the section of proofs.

Lemma 4.5.2 *For any true parameter $\theta\in\Theta$ and a sufficiently priced cycle l, on event (4.13) we have, for all $i\in\mathcal{N}$,*

$$\begin{aligned}
r(S,p,\theta) &\leq U_l(S,p),\\
U_l(S,p) &\leq r(S,p,\theta)+\frac{4\bar{p}(\overline{\kappa}-1)k_1\sqrt{\log(lN)}}{|S|(\underline{\kappa}-1)}\sum_{i\in S}||(1,-p_i)||_{\bar{V}_{i,l}^{-1}},
\end{aligned}$$

for any $S\in\mathcal{S}, p\in\mathcal{P}$.

Applying Lemma 4.5.2, we have that

$$\begin{aligned}
&r(S^*,p^*,\theta)-U_l(S^*,p^*)\leq 0\\
&U_l(S_l,p_l)-r(S_l,p_l,\theta)\leq\frac{4\bar{p}(\overline{\kappa}-1)k_1\sqrt{\log(lN)}}{|S|(\underline{\kappa}-1)}\sum_{i\in S}||(1,-p_i)||_{\bar{V}_{i,l}^{-1}}
\end{aligned}$$

on event (4.13) for all $i\in\mathcal{N}$, where k_1 is a constant defined in Proposition 4.5.1. Define the

event (4.13) as $\mathcal{E}_{\hat{\theta}_{i,l}}$, i.e, $\mathcal{E}_{\hat{\theta}_{i,l}} := \left\{||\hat{\theta}_{i,l} - \theta_i||_{\bar{V}_{i,l}} \leq k_1\sqrt{\log(8lN)}\right\}$. Then it follows from Proposition 4.5.1 that $\mathbb{P}(\mathcal{E}'_{\hat{\theta}_{i,l}}) \leq 1/(lN)$, and for event $\mathcal{E}_{\hat{\theta}_l} := \bigcap_{i\in\mathcal{N}} \mathcal{E}_{\hat{\theta}_{i,l}}$, we apply union bound to obtain $\mathbb{P}(\mathcal{E}'_{\hat{\theta}_l}) \leq 1/l$.

Now we are ready to derive the final Bayesian regret upper bound according to (4.22). For the first term in the last equation of (4.22), on event $\mathcal{E}_{\hat{\theta}_l}$, we have $r(S^*, p^*, \theta) \leq U_l(S^*, p^*)$. Thus,

$$\begin{aligned}\mathbb{E}\left[\sum_{l\in\mathcal{C}_1} |S_l|(r(S^*, p^*, \theta) - U_l(S^*, p^*))\right] =& \mathbb{E}\left[\sum_{l\in\mathcal{C}_1} |S_l|(r(S^*, p^*, \theta) - U_l(S^*, p^*))\mathbf{1}(\mathcal{E}_{\hat{\theta}_l})\right] \\ &+ \mathbb{E}\left[\sum_{l\in\mathcal{C}_1} |S_l|(r(S^*, p^*, \theta) - U_l(S^*, p^*))\mathbf{1}(\mathcal{E}'_{\hat{\theta}_l})\right] \\ \leq& \overline{p}K\sum_{l=1}^{L_T}\frac{1}{l} \leq \overline{p}K(\log(T)+1).\end{aligned}$$

For the second term in the last equation of (4.22), on event $\mathcal{E}_{\hat{\theta}_l}$ we have that for any fixed θ,

$$U_l(S_l, p_l) - r(S_l, p_l, \theta) \leq \frac{4\bar{p}k_1(\overline{\kappa}-1)\overline{\kappa}\sqrt{\log(lN)}}{|S_l|(\underline{\kappa}-1)}\sum_{i\in S_l}||z_{i,l}||_{\bar{V}_{i,l}^{-1}}.$$

By Lemma 11 in Abbasi-Yadkori et al. (2011), we have

$$\sum_{l\in\mathcal{T}_i(L_T+1)}||z_{i,l}||_{\bar{V}_{i,l}^{-1}} \leq 2\sqrt{Lt_i(L_T+1)\log(1+t_i(L_T+1)L/2)} \leq k_5\sqrt{t_i(L_T+1)\log(NT)}, \tag{4.23}$$

where k_5 is some constant chosen such that the second inequality above holds. Hence, we

have

$$
\begin{aligned}
&\mathbb{E}\left[\sum_{l\in\mathcal{C}_1}|S_l|(U_l(S_l,p_l)-r(S_l,p_l,\theta))\right]\\
=&\mathbb{E}\left[\sum_{l\in\mathcal{C}_1}|S_l|(U_l(S_l,p_l,\theta)-r(S_l,p_l))\mathbf{1}(\mathcal{E}_{\hat{\theta}_l})\right]+\mathbb{E}\left[\sum_{l\in\mathcal{C}_1}|S_l|(U_l(S_l,p_l,\theta)-r(S_l,p_l))\mathbf{1}(\mathcal{E}'_{\hat{\theta}_l})\right]\\
\leq& k_6\sqrt{\log(NT)}\mathbb{E}\left[\sum_{l\in\mathcal{C}_1}\sum_{i\in S_l}||z_{i,l}||_{\bar{V}_{i,l}^{-1}}\right]+\overline{p}K(\log T+1)\\
\leq& k_6\sqrt{\log(NT)}\mathbb{E}\left[\sum_{i\in\mathcal{N}}\sum_{l\in\mathcal{T}_i(L_T+1)}||z_{i,l}||_{\bar{V}_{i,l}^{-1}}\right]+\overline{p}K(\log T+1)\\
\leq& k_7\log(NT)\sum_{i\in\mathcal{N}}\sqrt{\mathbb{E}[t_i(L_T+1)]}+\overline{p}K(\log T+1)\\
=& k_7\log(NT)\sqrt{NT/(\underline{\kappa}-1)}+\overline{p}K(\log T+1),
\end{aligned}
$$

where $k_6=4\bar{p}k_2(\overline{\kappa}-1)\overline{\kappa}/(\underline{\kappa}-1)$, and $k_7=k_5k_6$ are some constants, the first inequality is from Lemma 4.5.2, the third inequality follows from (4.23) and Jensen's inequality, and the last equality is from (A.20) in Agrawal et al. (2017a). As a result, the Bayesian regret from sufficiently priced cycles can be bounded above by $O(\sqrt{NT}\log(NT)+K\log T)$.

Combining the Bayesian regrets for all three cases above yields the desired result, which completes the proof of Theorem 4.3.1.

4.6 Proofs of Technical Results

In this section, we provide all the missing proofs in the main body of the chapter, i.e., Proposition 4.5.1, Proposition 4.5.2, Corollary 4.5.1, Lemma 4.5.1, and Lemma 4.5.2.

4.6.1 Proof of Proposition 4.5.1

By definition, $\hat{\theta}_{i,l}$ is the maximizer of the likelihood function, i.e., the minimizer of the negative log-likelihood function within feasible region $\Theta_i:=[\underline{\alpha}_i,\overline{\alpha}_i]\times[\underline{\beta}_i,\overline{\beta}_i]$: $\hat{\theta}_{i,l}\in\arg\min_{\phi\in\Theta_i}\ L_{i,l}(\phi)$, where

$$
L_{i,l}(\phi)=\sum_{\tau\in\mathcal{T}_i(l)}\Big(-\hat{v}_{i,\tau}(x'_{i,\tau}\phi)+(1+\hat{v}_{i,\tau})\log(1+e^{x'_{i,\tau}\phi})\Big).
$$

By Taylor's theorem, we have for some $\check{\theta}_{i,l}$ on the line segment between θ_i and $\hat{\theta}_{i,l}$,

$$
\begin{aligned}
0 \geq & L_{i,l}(\hat{\theta}_{i,l}) - L_{i,l}(\theta_i) = \nabla L_{i,l}(\theta_i)'(\hat{\theta}_{i,l} - \theta_i) + \frac{1}{2}(\hat{\theta}_{i,l} - \theta_i)'\nabla^2 L_{i,l}(\check{\theta}_{i,l})(\hat{\theta}_{i,l} - \theta_i) \\
= & \left(\sum_{\tau \in \mathcal{T}_i(l)} \frac{-\hat{v}_{i,\tau} + e^{x'_{i,\tau}\theta_i}}{1 + e^{x'_{i,\tau}\theta_i}} x_{i,\tau} \right)' (\hat{\theta}_{i,l} - \theta_i) \\
& + \frac{1}{2}(\hat{\theta}_{i,l} - \theta_i)' \left(\sum_{\tau \in \mathcal{T}_i(l)} \frac{(1 + \hat{v}_{i,\tau}) e^{x'_{i,\tau}\check{\theta}_{i,l}}}{(1 + e^{x'_{i,\tau}\check{\theta}_{i,l}})^2} x_{i,\tau} x'_{i,\tau} \right) (\hat{\theta}_{i,l} - \theta_i) \\
\geq & - \left(\sum_{\tau \in \mathcal{T}_i(l)} \frac{\epsilon_{i,\tau} x_{i,\tau}}{1 + e^{x'_{i,\tau}\theta_i}} \right)' (\hat{\theta}_{i,l} - \theta_i) + \frac{(\underline{\kappa} - 1)\underline{\kappa}^2}{2\overline{\kappa}^2} (\hat{\theta}_{i,l} - \theta_i)' V_{i,l} (\hat{\theta}_{i,l} - \theta_i) \\
= & - \left(\sum_{\tau \in \mathcal{T}_i(l)} \frac{\epsilon_{i,\tau} x_{i,\tau}}{1 + e^{x'_{i,\tau}\theta_i}} \right)' (\hat{\theta}_{i,l} - \theta_i) + \frac{(\underline{\kappa} - 1)\underline{\kappa}^2}{2\overline{\kappa}^2} ||\hat{\theta}_{i,l} - \theta_i||^2_{\bar{V}_{i,l}} - \frac{(\underline{\kappa} - 1)\underline{\kappa}^2}{2\overline{\kappa}^2} ||\hat{\theta}_{i,l} - \theta_i||_2^2,
\end{aligned}
$$

where $V_{i,l} := \bar{V}_{i,l} - I = \sum_{\tau \in \mathcal{T}_i(l)} z_{i,\tau} z'_{i,\tau}$. Consequently, by Cauchy-Schwarz inequality,

$$
\frac{(\underline{\kappa} - 1)\underline{\kappa}^2}{2\overline{\kappa}^2} ||\hat{\theta}_{i,l} - \theta_i||^2_{\bar{V}_{i,l}} \leq \frac{(\underline{\kappa} - 1)\underline{\kappa}^2}{2\overline{\kappa}^2} ||\hat{\theta}_{i,l} - \theta_i||_2^2 + ||Z_{i,l}||_{\bar{V}_{i,l}^{-1}} ||\hat{\theta}_{i,l} - \theta_i||_{\bar{V}_{i,l}}.
$$

Now let us consider the event

$$
\mathcal{E}_{Z_{i,l}} := \{ ||Z_{i,l}||_{\bar{V}_{i,l}^{-1}} \leq \mu \sqrt{2\log(1 + t_i(l)L/2) + 2\log(k_2 l N)} \}
$$

as we highlighted in Section 4.5.1. Note that on this event, we have

$$
||\hat{\theta}_{i,l} - \theta_i||^2_{\bar{V}_{i,l}} \leq (\overline{\alpha} - \underline{\alpha})^2 + (\overline{\beta} - \underline{\beta})^2 + \frac{2\mu\overline{\kappa}^2}{(\underline{\kappa} - 1)\underline{\kappa}^2} \sqrt{2\log(1 + t_i(l)L/2) + 2\log(k_2 l N)} ||\hat{\theta}_{i,l} - \theta_i||_{\bar{V}_{i,l}}.
$$

This implies

$$
\begin{aligned}
||\hat{\theta}_{i,l} - \theta_i||_{\bar{V}_{i,l}} \leq & \sqrt{(\overline{\alpha} - \underline{\alpha})^2 + (\overline{\beta} - \underline{\beta})^2} + \frac{2\mu\overline{\kappa}^2}{(\underline{\kappa} - 1)\underline{\kappa}^2} \sqrt{2\log(1 + t_i(l)L/2) + 2\log(k_2 l N)} \\
\leq & k_1 \sqrt{\log(lN)},
\end{aligned}
$$

where the second inequality is obtained by choosing appropriate constant k_1.

As a result, we have

$$\begin{aligned}
&\mathbb{P}\left(||\hat{\theta}_{i,l}-\theta_i||_{\bar{V}_{i,l}} > k_1\sqrt{\log(lN)}\right)\\
=&\mathbb{P}\left(||\hat{\theta}_{i,l}-\theta_i||_{\bar{V}_{i,l}} > k_1\sqrt{\log(lN)}, \mathcal{E}_{Z_{i,l}}\right) + \mathbb{P}\left(||\hat{\theta}_{i,l}-\theta_i||_{\bar{V}_{i,l}} > k_1\sqrt{\log(lN)}, \mathcal{E}'_{Z_{i,l}}\right)\\
=&\mathbb{P}\left(||\hat{\theta}_{i,l}-\theta_i||_{\bar{V}_{i,l}} > k_1\sqrt{\log(lN)}, \mathcal{E}'_{Z_{i,l}}\right)\\
\leq&\mathbb{P}\left(\mathcal{E}'_{Z_{i,l}}\right)\\
=&\mathbb{P}\left(||Z_{i,l}||_{\bar{V}_{i,l}^{-1}} > \mu\sqrt{2\log(1+t_i(l)L/2)+2\log\left(k_2 lN\right)}\right),
\end{aligned}$$

where the notation $\mathcal{A}'$ is the complement of event $\mathcal{A}$. Then all remaining is to prove

$$\mathbb{P}\left(||Z_{i,l}||_{\bar{V}_{i,l}^{-1}} > \mu\sqrt{2\log(1+t_i(l)L/2)+2\log\left(k_2 lN\right)}\right) \leq 1/(lN).$$

To prove the above inequality, we need another result that is stated below, whose proof is postponed to the end of this section of proofs.

Lemma 4.6.1 *Let l be an arbitrary cycle with $t_i(l) \geq c_1\log(8lN)$, where*

$$c_1 \geq \max\{1, 1/\overline{p}^4\}4\underline{\kappa}^4\Lambda^2/\mu^2$$

. If $c_2 \geq 4L/(\mu\Lambda)$, then

$$||Z_{i,l}||_2 \leq c_2\mu^2\Lambda/(2\sqrt{L})\sqrt{t_i(l)\log(8lN)}$$

holds with probability at least $1-1/(2lN)$.

By Lemma 4.6.1, when $c_2 \geq 4L/(\mu\Lambda)$, in a sufficiently priced loop l we must have that $\mathbb{P}(\mathcal{E}'_{i,l}) < 1/(2lN)$ (recall that $\mathcal{E}_{i,l}$ is defined in (4.19)) because by definition of sufficiently priced loop,

$$\lambda_{\min}(\bar{V}_{i,l}) \geq \lambda_{\min}(W_{i,l}) \geq c_2\sqrt{t_i(l)\log(8lN)}.$$

Therefore,

$$\begin{aligned}
&\mathbb{P}\left(||Z_{i,l}||_{\bar{V}_{i,l}^{-1}} > \mu\sqrt{2\log(1+t_i(l)L/2)+2\log\left(k_2 lN\right)}\right)\\
=&\mathbb{P}\left(||Z_{i,l}||_{\bar{V}_{i,l}^{-1}} > \mu\sqrt{2\log(1+t_i(l)L/2)+2\log\left(k_2 lN\right)}, \mathcal{E}_{i,l}\right)\\
&+\mathbb{P}\left(||Z_{i,l}||_{\bar{V}_{i,l}^{-1}} > \mu\sqrt{2\log(1+t_i(l)L/2)+2\log\left(k_2 lN\right)}, \mathcal{E}'_{i,l}\right)\\
\leq&1/(2lN)+\mathbb{P}(\mathcal{E}'_{i,l})\\
\leq&1/(lN),
\end{aligned}$$

where the inequalities follow from Corollary 4.5.2 and Lemma 4.6.1. This completes the proof.

4.6.2 Proof of Proposition 4.5.2

We adopt an argument similar to that of Theorem 1 in Abbasi-Yadkori et al. (2011). First, we define

$$M_t^\sigma =: \exp\left(\sum_{s=1}^{t-1}\left[\frac{\epsilon_s\langle\sigma, z_s\rangle}{\mu} - \frac{\langle\sigma, z_s\rangle^2}{2}\right]\right) = \exp(\langle\sigma, \bar{Z}_t\rangle - ||\sigma||_{V_t}^2/2),$$

where $\bar{Z}_t := Z_t/\mu$ and σ is a vector chosen such that $||\sigma||_2 \cdot ||z_s||_2/\mu \leq \Lambda$ (by choosing $||\sigma||_2 \leq \mu\Lambda/\sqrt{L}$). It is not difficult to see that $\mathbb{E}[M_t^\sigma] \leq 1$ following the same argument as in Abbasi-Yadkori et al. (2011) and the fact that ϵ_s is sub-exponential with parameters (μ, Λ). Specifically, let

$$D_t^\sigma := \exp\left(\frac{\epsilon_s\langle\sigma, z_s\rangle}{\mu} - \frac{\langle\sigma, z_s\rangle^2}{2}\right).$$

By the definition of sub-exponential distribution and our choice of σ, we have $\mathbb{E}[D_t^\sigma|\mathcal{F}_{t-1}] \leq 1$. Furthermore,

$$\mathbb{E}[M_t^\sigma|\mathcal{F}_{t-2}] = \mathbb{E}[M_1^\sigma \cdots D_{t-2}^\sigma D_{t-1}^\sigma|\mathcal{F}_{t-2}] = D_1^\sigma \cdots D_{t-2}^\sigma \mathbb{E}[D_{t-1}^\sigma|\mathcal{F}_{t-2}] \leq M_{t-1}^\sigma.$$

This shows that M_t^σ is supermartingale with $\mathbb{E}[M_t^\sigma] \leq 1$. For a stopping time τ, we can follow Abbasi-Yadkori et al. (2011) to show that M_τ^σ is almost surely well-defined with $\mathbb{E}[M_\tau^\sigma] \leq 1$. By the convergence theorem for nonnegative supermartingales, $M_\infty^\sigma = \lim_{t\to\infty} M_t^\sigma$ is almost surely well-defined. Hence, M_τ^σ is well-defined regardless of $\tau < \infty$ or not. To see $\mathbb{E}[M_\tau^\sigma] \leq 1$, we let $Q_t^\sigma = M_{\min\{\tau,t\}}^\sigma$ be a stopped version of $(M_t^\sigma)_t$. Then by Fatou's Lemma, we have $\mathbb{E}[M_\tau^\sigma] = \mathbb{E}[\liminf_{t\to\infty} Q_t^\sigma] \leq \liminf_{t\to\infty} \mathbb{E}[Q_t^\sigma] \leq 1$.

Now, let Σ be a truncated multivariate normal random variable with covariance matrix V^{-1} which is truncated within $||\sigma||_2 \leq \mu\Lambda/\sqrt{L}$ and it is independent of all other random variables. Define $M_\tau = \mathbb{E}[M_\tau^\Sigma|\mathcal{F}_\infty]$, where the expectation is taken over σ and $\mathcal{F}_\infty$ is the tail σ-algebra of the filtration. Then by the definition of conditional expectation, $\mathbb{E}[M_\tau] = \mathbb{E}[\mathbb{E}[M_\tau^\Sigma|\Sigma]] \leq 1$.

We then present a lower bound for M_τ on $\mathcal{E}_\tau$. Let $f(\sigma)$ denote the density function of Σ, and for a positive definite matrix P, define $c(P) = \sqrt{(2\pi)^d/\det(P)} = \int \exp(-x'Px/2)dx$.

Then it can be seen that

$$f(\sigma) = \frac{\exp(-||\sigma||_V^2/2)}{\int_{||\sigma||_2 \le \mu\Lambda/L} \exp(-||\sigma||_V^2/2)d\sigma} \ge \frac{\exp(-||\sigma||_V^2/2)}{\int_{\mathbb{R}^d} \exp(-||\sigma||_V^2/2)d\sigma} = \frac{\exp(-||\sigma||_V^2/2)}{c(V)}.$$

Thus, on the event $\mathcal{E}_\tau$, we have

$$\begin{aligned}
M_\tau =& \int_{||\sigma||_2 \le \mu\Lambda/\sqrt{L}} \exp(\langle \sigma, \bar{Z}_\tau \rangle - ||\sigma||_{V_\tau}^2/2) f(\sigma) d\sigma \\
=& \int_{||\sigma||_2 \le \mu\Lambda/\sqrt{L}} \exp(-||\sigma - V_\tau^{-1}\bar{Z}_\tau||_{V_\tau}^2/2 + ||\bar{Z}_\tau||_{V_\tau^{-1}}^2/2) f(\sigma) d\sigma \\
\ge& \frac{\exp(||\bar{Z}_\tau||_{V_\tau^{-1}}^2/2)}{c(V)} \int_{||\sigma||_2 \le \mu\Lambda/L} \exp(-||\sigma - V_\tau^{-1}Z_\tau||_{V_\tau}^2/2 - ||\sigma||_V^2/2) d\sigma.
\end{aligned}$$

It can be checked that

$$||\sigma - V_\tau^{-1}\bar{Z}_\tau||_{V_\tau}^2 + ||\sigma||_V^2 = ||\sigma - (V_\tau + V)^{-1}\bar{Z}_\tau||_{V+V_\tau}^2 + ||\bar{Z}_\tau||_{V_\tau^{-1}}^2 - ||\bar{Z}_\tau||_{(V+V_\tau)^{-1}}^2,$$

which implies

$$\begin{aligned}
M_\tau \ge& \frac{\exp(||\bar{Z}_\tau||_{(V+V_\tau)^{-1}}^2/2)}{c(V)} \int_{||\sigma||_2 \le \mu\Lambda/\sqrt{L}} \exp(-||\sigma - (V_\tau + V)^{-1}\bar{Z}_\tau||_{V+V_\tau}^2/2) d\sigma \\
\ge& \frac{\exp(||\bar{Z}_\tau||_{(V+V_\tau)^{-1}}^2/2)}{c(V)} \int_{||\sigma||_2 \le \mu\Lambda/(2\sqrt{L})} \exp(-||\sigma||_{V+V_\tau}^2/2) d\sigma,
\end{aligned}$$

where the second inequality is due to that fact that on $\mathcal{E}_\tau$, $||(V_\tau + V)^{-1}\bar{Z}_\tau||_2 \le \mu\Lambda/(2\sqrt{L})$. Note that

$$\begin{aligned}
& \int_{||\sigma||_2 \le \mu\Lambda/(2\sqrt{L})} \exp(-||\sigma||_{V+V_\tau}^2/2) d\sigma \\
=& c(V+V_\tau) \int_{||\sigma||_2 \le \mu\Lambda/(2\sqrt{L})} \exp(-||\sigma||_{V+V_\tau}^2/2)/c(V+V_\tau) d\sigma \\
=& c(V+V_\tau) \int_{||\sigma||_{(V+V_\tau)^{-1}} \le \mu\Lambda/(2\sqrt{L})} \exp(-||\sigma||_2^2/2)/c(I) d\sigma \\
\ge& c(V+V_\tau) \int_{||\sigma||_{V^{-1}} \le \mu\Lambda/(2\sqrt{L})} \exp(-||\sigma||_2^2/2)/c(I) d\sigma \\
=& k(\mu, \Lambda, L, V) c(V+V_\tau),
\end{aligned}$$

where the second equality follows from change of variable, and the first inequality is by $V + V_\tau \succeq V$ (recall that $A \succeq B$ if $A - B$ is a semi-definite matrix), and $k(\mu, \Lambda, L, V) \in$

$(0,1)$ is a constant given by

$$k(\mu,\Lambda,L,V) := \int_{||\sigma||_{V^{-1}} \leq \mu\Lambda/(2\sqrt{L})} \exp(-||\sigma||_2^2/2)/c(I)d\sigma.$$

Putting things together, we get that

$$\begin{aligned} M_\tau \geq & k(\mu,\Lambda,L,V)\frac{c(V+V_\tau)}{c(V)}\exp(||\bar{Z}_\tau||^2_{(V+V_\tau)^{-1}}/2) \\ = & k(\mu,\Lambda,L,V)\left(\frac{\det(V)}{\det(V+V_\tau)}\right)^{1/2}\exp(||\bar{Z}_\tau||^2_{(V+V_\tau)^{-1}}/2). \end{aligned}$$

Finally, we obtain that

$$\begin{aligned} & \mathbb{P}\left(||\bar{Z}_\tau||^2_{\bar{V}_\tau^{-1}} > 2\log\left(\frac{\det(\bar{V}_\tau)^{1/2}}{k(\mu,\Lambda,L,V)\delta\det(V)^{1/2}}\right), \mathcal{E}_\tau\right) \\ = & \mathbb{P}\left(\frac{\exp(||\bar{Z}_\tau||^2_{\bar{V}_\tau^{-1}}/2)k(\mu,\Lambda,L,V)}{\delta^{-1}(\det(\bar{V}_\tau)/\det(V))^{1/2}} > 1, \mathcal{E}_\tau\right) \\ \leq & \mathbb{E}\left[\frac{\exp(||\bar{Z}_\tau||_{\bar{V}_\tau^{-1}}/2)k(\mu,\Lambda,L,V)}{\delta^{-1}(\det(\bar{V}_\tau)/\det(V))^{1/2}}\mathbf{1}(\mathcal{E}_\tau)\right] \leq \mathbb{E}[M_\tau\mathbf{1}(\mathcal{E}_\tau)]\delta \leq \delta, \end{aligned}$$

where the first inequality is by Markov inequality. The proof of Proposition 4.5.2 is thus complete.

4.6.3 Proof of Corollary 4.5.1

Let λ_i, $i = 1,\ldots,d$, be the eigenvalues of $\bar{V}_\tau$, that are positive because $\bar{V}_\tau$ is positive definite. By the inequality of arithmetic and geometric means,

$$\begin{aligned} \left(\prod_{i=1}^d \lambda_i\right)^{1/d} \leq \frac{\sum_{i=1}^d \lambda_i}{d} = \frac{\text{trace}(\bar{V}_\tau)}{d} &= \frac{\text{trace}(I)}{d} + \sum_{t=1}^\tau \frac{\text{trace}(z_t z_t')}{d} \\ &= 1 + \sum_{t=1}^\tau ||z_t||_2^2/d \leq 1 + \tau L/d. \end{aligned}$$

Thus,

$$\frac{\det(\bar{V}_t)}{\det V} \leq \ (1+\tau L/d)^d.$$

Combining with the results in Proposition 4.5.2 proves the desired result.

4.6.4 Proof of Lemma 4.5.1

To prove this lemma, we first establish a result on the minimum eigenvalue of the expected matrix: For any product $i \in \mathcal{N}$ and cycle l, let price $p_{i,l} = p'_{i,l} + \omega_{i,l}$, where $p'_{i,l} \in [\underline{p}_i + |\omega_{i,l}|, \overline{p}_i - |\omega_{i,l}|]$ is an arbitrary price, and $\omega_{i,l}$ is a random variable taking value $\pm|\omega_{i,l}|$ with equal probability. Then we have $\lambda_{\min}\big(\mathbb{E}[\underline{z}_{i,l}\underline{z}'_{i,l}|\mathcal{F}_{l-1}]\big) \geq \omega_{i,l}^2/(\overline{\kappa}^2\underline{\kappa}^2 L) > 0$.

The Fisher's information matrix for our problem can be written as

$$\mathbb{E}[\underline{z}_{i,l}\underline{z}'_{i,l}|\mathcal{F}_{l-1}] \geq \frac{1}{\overline{\kappa}^2}\mathbb{E}[x_{i,l}x'_{i,l}|\mathcal{F}_{l-1}] = \frac{1}{\overline{\kappa}^2}\begin{bmatrix} 1 & -p'_{i,l} \\ -p'_{i,l} & (p'_{i,l})^2 + \omega_{i,l}^2 \end{bmatrix}.$$

The minimum eigenvalue of this 2×2 matrix can be easily found as

$$\begin{aligned} &\lambda_{\min}(\mathbb{E}[x_{i,l}x'_{i,l}|\mathcal{F}_{l-1}]) \\ =&\frac{((p'_{i,l})^2 + \omega_{i,l}^2 + 1) - ((p'_{i,l})^2 + \omega_{i,l}^2 + 1)\sqrt{1 - 4\omega_{i,l}^2/((p'_{i,l})^2 + \omega_{i,l}^2 + 1)^2}}{2}. \end{aligned}$$

Simple algebra shows that

$$\lambda_{\min}(\mathbb{E}[x_{i,l}x'_{i,l}|\mathcal{F}_{l-1}]) \geq \frac{\omega_{i,l}^2}{(p'_{i,l})^2 + \omega_{i,l}^2 + 1} \geq \frac{\omega_{i,l}^2}{\overline{p}_i^2 + 1} \geq \frac{\omega_{i,l}^2}{L\underline{\kappa}^2}.$$

Applying this result, we obtain $\lambda_{\min}(\mathbb{E}[\underline{z}_{i,l}\underline{z}'_{i,l}|\mathcal{F}_{l-1}]) \geq \omega_{i,l}^2/(\overline{\kappa}^2\underline{\kappa}^2 L)$. As a result,

$$\lambda_{\min}\Big(\sum_{\tau\in\mathcal{T}_i(l)} \mathbb{E}[\underline{z}_{i,\tau}\underline{z}'_{i,\tau}|\mathcal{F}_{\tau-1}]\Big) \geq \sum_{\tau\in\mathcal{T}_i(l)} \lambda_{\min}(\mathbb{E}[\underline{z}_{i,\tau}\underline{z}'_{i,\tau}|\mathcal{F}_{\tau-1}]) \geq \tilde{t}_i(l)\omega_0^2/(\overline{\kappa}^2\underline{\kappa}^2 L).$$

Therefore, we have that

$$
\begin{aligned}
&\mathbb{P}\left(\lambda_{\min}\left(\sum_{\tau\in\mathcal{T}_i(l)} \underline{z}_{i,\tau}\underline{z}'_{i,\tau}\right) < \frac{\tilde{t}_i(l)\omega_0^2}{2\overline{\kappa}^2\underline{\kappa}^2 L}\right)\\
=&\mathbb{P}\left(\lambda_{\min}\left(\sum_{\tau\in\mathcal{T}_i(l)} \underline{z}_{i,\tau}\underline{z}'_{i,\tau}\right) < \frac{\tilde{t}_i(l)\omega_0^2}{2\overline{\kappa}^2\underline{\kappa}^2 L}, \lambda_{\min}\left(\sum_{\tau\in\mathcal{T}_i(l)} \mathbb{E}[\underline{z}_{i,\tau}\underline{z}'_{i,\tau}|\mathcal{F}_{\tau-1}]\right) \geq \frac{\tilde{t}_i(l)\omega_0^2}{\overline{\kappa}^2\underline{\kappa}^2 L}\right)\\
\leq &\sum_{k=\lceil c_1\log(8lN)\rceil}^{l} \mathbb{P}\left(\lambda_{\min}\left(\sum_{\tau\in\mathcal{T}_i(l)} \underline{z}_{i,\tau}\underline{z}'_{i,\tau}\right) < \frac{k\omega_0^2}{2\overline{\kappa}^2\underline{\kappa}^2 L},\right.\\
&\qquad\left.\lambda_{\min}\left(\sum_{\tau\in\mathcal{T}_i(l)} \mathbb{E}[\underline{z}_{i,\tau}\underline{z}'_{i,\tau}|\mathcal{F}_{\tau-1}]\right) \geq \frac{k\omega_0^2}{\overline{\kappa}^2\underline{\kappa}^2 L}, \tilde{t}_i(l)=k\right)\\
< &\sum_{k=\lceil c_1\log(8lN)\rceil}^{l} 2\exp\left(-\frac{k\omega_0^2}{4\overline{\kappa}^2\underline{\kappa}^2 L^2}\right) \leq 1/(2lN),
\end{aligned}
$$

where the first inequality is due to $\tilde{t}_i(l) \in [c_1\log(8lN), l]$, the second inequality follows from Theorem 3.1 in Tropp (2011) with $\delta = 1/2$, and the last inequality is by our choice of c_1. The proof is complete.

4.6.5 Proof of Lemma 4.5.2

For any S, p and $i \in \mathcal{N}$ on event (4.13), Cauchy-Schwarz implies that

$$
|e^{\alpha_i-\beta_i p_i} - e^{\hat{\alpha}_{i,l}-\hat{\beta}_{i,l}p_i}| \leq (\overline{\kappa}-1)||\hat{\theta}_{i,l}-\theta_i||_{\bar{V}_{i,l}}||(1,-p_i)||_{\bar{V}_{i,l}^{-1}}.
$$

Applying Taylor's theorem, we obtain

$$
\begin{aligned}
|r(S,p,\theta) - r(S,p,\hat{\theta}_l)| &\leq \frac{2\bar{p}}{|S|(\underline{\kappa}-1)}\sum_{i\in S}|e^{\alpha_i-\beta_i p_i} - e^{\hat{\alpha}_{i,l}-\hat{\beta}_{i,l}p_i}|\\
&\leq \frac{2\bar{p}(\overline{\kappa}-1)}{|S|(\underline{\kappa}-1)}\sum_{i\in S}||\hat{\theta}_{i,l}-\theta_i||_{\bar{V}_{i,l}}||(1,-p_i)||_{\bar{V}_{i,l}^{-1}}.
\end{aligned}
$$

By the definition of $U_l(S,p)$ in (4.21), the inequality above implies (4.23) and

$$
U_l(S,p) \leq r(S,p,\theta) + \frac{4\bar{p}(\overline{\kappa}-1)}{|S|(\underline{\kappa}-1)}\sum_{i\in S}||\hat{\theta}_{i,l}-\theta_i||_{\bar{V}_{i,l}}||(1,-p_i)||_{\bar{V}_{i,l}^{-1}}.
$$

Since $||\hat{\theta}_{i,l} - \theta_i||_{\bar{V}_{i,l}} \leq k_1\sqrt{\log(lN)}$ from event (4.13) for all $i \in \mathcal{N}$, this completes the proof of Lemma 4.5.2.

4.6.6 Proof of Lemma 4.6.1

Note that

$$||Z_{i,l}||_2 = \sqrt{\Big(\sum_{\tau\in\mathcal{T}_i(l)} \epsilon_{i,\tau}/(1+e^{x'_{i,\tau}\theta_i})\Big)^2 + \Big(\sum_{\tau\in\mathcal{T}_i(l)} \epsilon_{i,\tau}p_{i,\tau}/(1+e^{x'_{i,\tau}\theta_i})\Big)^2},$$

so it suffices to bound $\left|\sum_\tau \epsilon_{i,\tau}/(1+\exp(x'_{i,\tau}\theta_i))\right|$ and $\left|\sum_\tau \epsilon_{i,\tau}p_{i,\tau}/(1+\exp(x'_{i,\tau}\theta_i))\right|$. Define $\tilde{\epsilon}_{i,\tau} := \epsilon_{i,\tau}/(1+\exp(x'_{i,\tau}\theta_i)$. Note that the sequence of $\epsilon_{i,\tau}$ is a martingale difference sequence with $\mathbb{E}[\exp(\lambda\epsilon_{i,\tau})|\mathcal{F}_{\tau-1}] \leq \exp(\mu^2\lambda^2/2)$ for all $|\lambda| \leq \Lambda$ and τ. As a result, $\tilde{\epsilon}_{i,\tau}$ is also a martingale difference sequence such that $\mathbb{E}[\exp(\lambda\tilde{\epsilon}_{i,\tau})|\mathcal{F}_{\tau-1}] \leq \exp(\tilde{\mu}^2\lambda^2/2)$ for all $|\lambda| \leq \tilde{\Lambda}$, where $\tilde{\mu} = \mu/\underline{\kappa}$ and $\tilde{\Lambda} = \underline{\kappa}\Lambda$.

On the range of $t_i(l)$, we have

$$\begin{aligned}
&\mathbb{P}\left(\left|\sum_{\tau\in\mathcal{T}_i(l)} \tilde{\epsilon}_{i,\tau}\right| > 2\tilde{\mu}\sqrt{\log(8lN)t_i(l)}\right)\\
\leq &\sum_{k=\lceil c_1\log(8lN)\rceil}^{l} \mathbb{P}\left(\left|\sum_{\tau\in\mathcal{T}_i(l)} \tilde{\epsilon}_{i,\tau}\right| > 2\tilde{\mu}\sqrt{\log(8lN)k}, t_i(l) = k\right)\\
< &\sum_{k=\lceil c_1\log(8lN)\rceil}^{l} 2e^{-2\log(8lN)} \leq 1/(4lN),
\end{aligned}$$

where the first inequality follows from $t_i(l) \in [c_1\log(8lN), l]$, and the second inequality is from Theorem 15 in Petrov (2012) adapted to martingale difference sequence and the fact that $2\tilde{\mu}\sqrt{\log(8lN)k} \leq k\tilde{\mu}^2/\tilde{\Lambda}$ for all $k \geq c_1\log(8lN)$ with $c_1 \geq 4\tilde{\Lambda}^2/\tilde{\mu}^2$.

Similarly, we obtain bound

$$\mathbb{P}\left(\left|\sum_\tau \tilde{\epsilon}_{i,\tau}p_{i,\tau}\right| > 2\tilde{\mu}\sqrt{\log(8lN)t_i(l)\overline{p}^2}\right) < 1/(4lN)$$

when $t_i(l) \geq 4\tilde{\Lambda}^2\log(8lN)/(\tilde{\mu}^2\overline{p}^4)$.

Combining the two inequalities above together with the assumption $c_2 \geq 4L/(\mu\Lambda)$ completes the proof of the lemma.

4.7 Conclusion

With the rapid development in e-commerce, assortment optimization and pricing are receiving increasingly more attention from both academia and industry. Even though voluminous literature exists on pricing or assortment optimization separately, there exist few studies on their joint optimization. In this work we have studied a version of the joint assortment and pricing optimization problem where customer purchasing behavior is not completely known *a priori*, and developed a learning algorithm that maximizes the expected revenue on the fly. To our best knowledge, this is the first work on joint dynamic assortment optimization and pricing problem with no prior information about customer demand.

The multinomial logit (MNL) model is employed in this chapter but customer choice parameters are unknown. We develop a learning algorithm that adaptively updates these parameters in carefully designed cycles. Our algorithm is based on Thompson Sampling in choosing parameters to avoid complicated optimization procedure (see the discussion of equation (4.12)) in each stage and enforces product testing and price perturbation. We show that its Bayesian regret is bounded above by $O(N\log(NT)+\sqrt{NT}\log(NT))$ which is independent of problem instance. Numerical experiments are conducted and the results show that the algorithm performs very well and in particular, it outperforms the benchmark algorithms. Thus, both the theoretical result and numerical result indicate that the TS-PS algorithm can be an effective method for solving practical dynamic joint assortment and pricing optimization problem, and in particular, they suggest that simultaneously exploring and learning about the optimal assortment and pricing decisions is preferable over sequentially finding the optimal assortment and then restricting the price exploration to the latter subset.

Chapter 5

Summary and Conclusion

The overall objective of this thesis is to develop data-driven algorithms for dynamic decision making problems in revenue management with unknown demand. In particular, we tackle three different problems which are prevalent in industry: dynamic pricing, personalized assortment optimization, and joint dynamic pricing and assortment optimization. The following is a summary of the most important findings of Chapters 2, 3, and 4, corresponding to these three problems, respectively.

Chapter 2 studied the problem of dynamic pricing for products with low sales and popularity. Based on the idea of product clustering, two learning algorithms were developed in this chapter: one for a dynamic pricing problem with the generalized linear demand, and another for the special case of linear demand functions under weaker assumptions on product covariates. Regrets of both algorithms were established under mild technical conditions. We tested our algorithms on a real dataset from Alibaba Group by simulating the demand function. Results showed that in all numerical experiments, both algorithms outperformed several benchmarks, where one considered all products separately, and another one treated all products as a single cluster. A field experiment was conducted at Alibaba by implementing the CSMP algorithm on a set of products, and the results showed that our algorithm could significantly increase revenue.

In Chapter 3, we presented several algorithms for personalized assortment optimization where customer's choice model follows from MNL. In this study, we designed two adaptive algorithms that learn the demand on the fly. The first one, P-UCB, uses MLE for parameter estimation and applies personalized UCB for assortment optimization in demand exploration. The second algorithm, OLP-UCB, bears similar structure as P-UCB but applies an online convex optimization scheme for parameter optimization. OLP-UCB has a constant computational time (in contrast to linearly increasing time of P-UCB) in each iteration, so it significantly reduces computational cost when large historical data has been collected. We proved that OLP-UCB, with significant improvement in computational complexity, achieved similar performance as P-UCB both theoretically and numerically, hence

addressing the challenge of large number of data samples. We then considered the online personalized assortment optimization problem with high dimensional customers' data. To tackle the data high dimensionality challenge, we applied random projection method to reduce the dimension for the sake of accelerated computation. The theoretical and numerical performances of the developed algorithm OLP-UCB-RP were proved to be promising given sparsity structure of customers' data.

In Chapter 4, we investigated the dynamic joint pricing and assortment optimization. Customer's choice again follows from MNL model with unknown parameters. A learning algorithms was developed based on a modification of Thompson sampling. More specifically, we divided the time horizon into carefully designed cycles, and divided these cycles into cycles of forced testing of prices and assortments, and the ones of applying Thompson sampling. Theoretical performance upper bound was proved to be promising, and numerical experiments based on synthetic data were conducted and results showed that our algorithm outperformed several important benchmarks.

There are several directions for future research. The first one is to investigate the performance of dynamic pricing with product clustering in a more general setting. Chapter 2 essentially assumes a natural clustering structure existing among all products. In a more general setting, for instance, we can investigate the case that the parameters of each product are generated from a common prior distribution. Moreover, we can study the best strategy of clustering as individual data of each product ranges from very scarce to very abundant. The second direction of future research is to develop data-driven algorithms for personalized assortment optimization with more general choice models than MNL. For instance, when customer choice model is from nested logit (NL) or any nonparametric model, one will need to modify the algorithm to adapt to that specific setting. The third direction is a further study of the joint pricing and assortment optimization problem. Our current algorithm is based on Thompson sampling, which is a randomized policy. While in reality, sometimes the decision maker prefers a deterministic policy so further investigation is needed for such a policy with comparable theoretical and numerical performance.

Bibliography

Abbasi-Yadkori Y, Pál D, Szepesvári C (2011) Improved algorithms for linear stochastic bandits. *Advances in Neural Information Processing Systems*, 2312–2320.

Abbasi-Yadkori Y, Pál D, Szepesvári C (2012) Online-to-confidence-set conversions and application to sparse stochastic bandits. *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, 1–9.

Agrawal S, Avadhanula V, Goyal V, Zeevi A (2017a) MNL-bandit: a dynamic learning approach to assortment selection. *arXiv preprint arXiv:1706.03880*.

Agrawal S, Avadhanula V, Goyal V, Zeevi A (2017b) Thompson sampling for the MNL-bandit. *arXiv preprint arXiv:1706.00977*.

Agrawal S, Goyal N (2013) Thompson sampling for contextual bandits with linear payoffs. *International Conference on Machine Learning*, 127–135.

Akday Y, Natarajan H, Xu S (2010) Joint dynamic pricing of multiple perishable products under consumer choice. *Management Science*, 56(8):1345–1361.

Alptekinoğlu A, Semple JH (2016) The exponomial choice model: A new alternative for assortment and price optimization. *Operations Research*, 64(1):79–93.

Andrieu C, De Freitas N, Doucet A, Jordan MI (2003) An introduction to MCMC for machine learning. *Machine Learning*, 50(1-2):5–43.

Aouad A, Levi R, Segev D (2018) Greedy-like algorithms for dynamic assortment planning under multinomial logit preferences. *Operations Research*, 66(5):1321–1345.

Araman VF, Caldentey R (2009) Dynamic pricing for nonperishable products with demand learning. *Operations Research*, 57(5):1169–1188.

Arthur D, Vassilvitskii S (2007) k-means++: The advantages of careful seeding. *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, 1027–1035.

Auer P (2002) Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422.

Auer P, Cesa-Bianchi N, Fischer P (2002) Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256.

Baardman L, Levin I, Perakis G, Singhvi D (2017) Leveraging comparables for new product sales forecasting. *Available at SSRN 3086237*.

Bahmani B, Moseley B, Vattani A, Kumar R, Vassilvitskii S (2012) Scalable k-means++. *Proceedings of the VLDB Endowment*, 5(7):622–633.

Ban GY, Gallien J, Mersereau AJ (2018) Dynamic procurement of new products with covariate information: The residual tree method. *Manufacturing & Service Operations Management*.

Ban GY, Keskin NB (2017) Personalized dynamic pricing with machine learning. *Available at SSRN 2972985*.

Bastani H, Bayati M (2015) Online decision-making with high-dimensional covariates. *Available at SSRN 2661896*.

Bastani H, Simchi-Levi D, Zhu R (2019) Meta dynamic pricing: Learning across experiments. *Available at SSRN 3334629*.

Ben-Akiva ME, Lerman SR (1985) *Discrete choice analysis: theory and application to travel demand*, volume 9 (MIT press).

Bernstein F, Kök AG, Xie L (2015) Dynamic assortment customization with limited inventories. *Manufacturing & Service Operations Management*, 17(4):538–553.

Bernstein F, Modaresi S, Sauré D (2018) A dynamic clustering approach to data-driven assortment personalization. *Management Science*, 65(5):2095–2115.

Bertsimas D, Perakis G (2006) Dynamic pricing: A learning approach. *Mathematical and Computational Models for Congestion Charging*, 45–79 (Springer).

Besbes O, Gur Y, Zeevi A (2015) Non-stationary stochastic optimization. *Operations Research*, 63(5):1227–1244.

Besbes O, Sauré D (2016) Product assortment and price competition under multinomial logit demand. *Production and Operations Management*, 25(1):114–127.

Besbes O, Zeevi A (2009) Dynamic pricing without knowing the demand function: Risk bounds and near-optimal algorithms. *Operations Research*, 57(6):1407–1420.

Besbes O, Zeevi A (2012) Blind network revenue management. *Operations Research*, 60(6):1537–1550.

Besbes O, Zeevi A (2015) On the (surprising) sufficiency of linear models for dynamic pricing with demand learning. *Management Science*, 61(4):723–739.

Bezdek JC (2013) *Pattern recognition with fuzzy objective function algorithms* (Springer Science & Business Media).

Bitran G, Caldentey R (2003) An overview of pricing models for revenue management. *Manufacturing & Service Operations Management*, 5(3):203–229.

Bobadilla J, Ortega F, Hernando A, Gutiérrez A (2013) Recommender systems survey. *Knowledge-Based Systems*, 46:109–132.

Broder J, Rusmevichientong P (2012) Dynamic pricing under a general parametric choice model. *Operations Research*, 60(4):965–980.

Brown RG (1959) *Statistical forecasting for inventory control* (McGraw/Hill).

Bubeck S, Cesa-Bianchi N, et al. (2012) Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122.

Bubeck S, Eldan R, Lee YT (2016) Kernel-based methods for bandit convex optimization. *arXiv preprint arXiv:1607.03084*.

Bubeck S, Liu CY (2013) Prior-free and prior-dependent regret bounds for thompson sampling. *Advances in Neural Information Processing Systems*, 638–646.

Cachon GP, Kök AG (2007) Category management and coordination in retail assortment planning in the presence of basket shopping consumers. *Management Science*, 53(6):934–951.

Caro F, Gallien J (2007) Dynamic assortment with demand learning for seasonal consumer goods. *Management Science*, 53(2):276–292.

Carpentier A, Munos R (2012) Bandit theory meets compressed sensing for high dimensional stochastic linear bandit. *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, 190–198.

Carvalho AX, Puterman ML (2005) Learning and pricing in an internet environment with binomial demands. *Journal of Revenue and Pricing Management*, 3(4):320–336.

Cesa-Bianchi N, Gentile C, Zappella G (2013) A gang of bandits. *Advances in Neural Information Processing Systems*, 737–745.

Chang PC, Wang YW, Tsai CY (2005) Evolving neural network for printed circuit board sales forecasting. *Expert Systems with Applications*, 29(1):83–92.

Chapelle O, Li L (2011) An empirical evaluation of Thompson sampling. *Advances in Neural Information Processing Systems*, 2249–2257.

Chen KD, Hausman WH (2000) Technical note: Mathematical properties of the optimal product line selection problem using choice-based conjoint analysis. *Management Science*, 46(2):327–332.

Chen M, Chen ZL (2015) Recent developments in dynamic pricing research: multiple products, competition, and limited demand information. *Production and Operations Management*, 24(5):704–731.

Chen N, Gallego G (2018) Nonparametric learning and optimization with covariates. *Available at SSRN 3172697*.

Chen Q, Jasin S, Duenyas I (2015a) Real-time dynamic pricing with minimal and flexible price adjustment. *Management Science*, 62(8):2437–2455.

Chen R, Jiang H (2017) Capacitated assortment and price optimization for customers with disjoint consideration sets. *Operations Research Letters*, 45(2):170–174.

Chen X, Owen Z, Pixton C, Simchi-Levi D (2015b) A statistical learning approach to personalization in revenue management. *Available at SSRN 2579462*.

Chen X, Wang Y (2017) A note on tight lower bound for MNL-bandit assortment selection models. *arXiv preprint arXiv:1709.06109*.

Chen X, Wang Y, Zhou Y (2018a) Dynamic assortment optimization with changing contextual information. *arXiv preprint arXiv:1810.13069*.

Chen X, Wang Y, Zhou Y (2018b) Dynamic assortment selection under the nested logit models. *arXiv preprint arXiv:1806.10410*.

Chen Y, Shi C (2019) Network revenue management with online inverse batch gradient descent method. *Available at SSRN 3331939*.

Cheung WC, Simchi-Levi D (2017) Thompson sampling for online personalized assortment optimization problems with multinomial logit choice models. *Available at SSRN 3075658*.

Cheung WC, Simchi-Levi D, Wang H (2017) Dynamic pricing and demand learning with limited price experimentation. *Operations Research*, 65(6):1722–1731.

Chu W, Li L, Reyzin L, Schapire R (2011) Contextual bandits with linear payoff functions. *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 208–214.

Covington P, Adams J, Sargin E (2016) Deep neural networks for youtube recommendations. *Proceedings of the 10th ACM Conference on Recommender Systems*, 191–198 (ACM).

Cremonesi P, Koren Y, Turrin R (2010) Performance of recommender algorithms on top-n recommendation tasks. *Proceedings of the Fourth ACM Conference on Recommender Systems*, 39–46 (ACM).

Cross RG (1995) An introduction to revenue management. *Handbook of Airline Economics*.

Dani V, Hayes TP, Kakade SM (2008) Stochastic linear optimization under bandit feedback. *21st Annual Conference on Learning Theory*, 355–366.

Davidson J, Liebald B, Liu J, Nandy P, Van Vleet T, Gargi U, Gupta S, He Y, Lambert M, Livingston B, et al. (2010) The youtube video recommendation system. *Proceedings of the Fourth ACM Conference on Recommender Systems*, 293–296 (ACM).

Davis J, Gallego G, Topaloglu H (2013) Assortment planning under the multinomial logit model with totally unimodular constraint structures. *Working Paper*.

Davis JM, Gallego G, Topaloglu H (2014) Assortment optimization under variants of the nested logit model. *Operations Research*, 62(2):250–273.

den Boer AV (2014) Dynamic pricing with multiple products and partially specified demand distribution. *Mathematics of Operations Research*, 39(3):863–888.

den Boer AV (2015) Dynamic pricing and learning: historical origins, current research, and new directions. *Surveys in Operations Research and Management Science*, 20(1):1–18.

den Boer AV, Zwart B (2013) Simultaneously learning and optimizing using controlled variance pricing. *Management Science*, 60(3):770–783.

Dong L, Kouvelis P, Tian Z (2009) Dynamic pricing and inventory control of substitute products. *Manufacturing & Service Operations Management*, 11(2):317–339.

Dunn JC (1973) A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(3):32–57.

Elmaghraby W, Keskinocak P (2003) Dynamic pricing in the presence of inventory considerations: Research overview, current practices, and future directions. *Management Science*, 49(10):1287–1309.

Fan X, Grama I, Liu Q, et al. (2015) Exponential inequalities for martingales with applications. *Electronic Journal of Probability*, 20(1):22.

Farias VF, Van Roy B (2010) Dynamic pricing with a prior on market response. *Operations Research*, 58(1):16–29.

Feldman J, Zhang D, Liu X, Zhang N (2018) Taking assortment optimization from theory to practice: Evidence from large field experiments on alibaba. *Available at SSRN 3232059*.

Ferreira KJ, Lee BHA, Simchi-Levi D (2015) Analytics for an online retailer: Demand forecasting and price optimization. *Manufacturing & Service Operations Management*, 18(1):69–88.

Ferreira KJ, Simchi-Levi D, Wang H (2018a) Online network revenue management using thompson sampling. *Operations Research*, 66(6):1586–1602.

Ferreira KJ, Simchi-Levi D, Wang H (2018b) Online network revenue management using Thompson sampling. *Forthcoming at Operations Research*.

Filippi S, Cappe O, Garivier A, Szepesvári C (2010) Parametric bandits: The generalized linear case. *Advances in Neural Information Processing Systems*, 586–594.

Frazier PI, Wang J (2016) Bayesian optimization for materials design. Lookman T, Alexander FJ, Rajan K, eds., *Information Science for Materials Discovery and Design*, 45–75 (Springer, Cham).

Gallego G, Li A, Truong VA, Wang X (2016) Online personalized resource allocation with customer choice. Technical report, Working Paper. http://arxiv. org/abs/1511.01837 v1.

Gallego G, Topaloglu H (2014) Constrained assortment optimization for the nested logit model. *Management Science*, 60(10):2583–2601.

Gallego G, Van Ryzin G (1994) Optimal dynamic pricing of inventories with stochastic demand over finite horizons. *Management Science*, 40(8):999–1020.

Gallego G, Van Ryzin G (1997) A multiproduct dynamic pricing problem and its applications to network yield management. *Operations Research*, 45(1):24–41.

Gallego G, Wang R (2014) Multiproduct price optimization and competition under the nested logit model with product-differentiated price sensitivities. *Operations Research*, 62(2):450–461.

Gaur V, Honhon D (2006) Assortment planning and inventory decisions under a locational choice model. *Management Science*, 52(10):1528–1543.

Gentile C, Li S, Kar P, Karatzoglou A, Etrue E, Zappella G (2016) On context-dependent clustering of bandits. *arXiv preprint arXiv:1608.03544*.

Gentile C, Li S, Zappella G (2014) Online clustering of bandits. *International Conference on Machine Learning*, 757–765.

Golrezaei N, Nazerzadeh H, Rusmevichientong P (2014) Real-time optimization of personalized assortments. *Management Science*, 60(6):1532–1551.

Gomez-Uribe CA, Hunt N (2016) The netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems (TMIS)*, 6(4):13.

Goyal V, Levi R, Segev D (2016) Near-optimal algorithms for the assortment planning problem under dynamic substitution and stochastic demand. *Operations Research*, 64(1):219–235.

Hanson W, Martin K (1996) Optimizing multinomial logit profit functions. *Management Science*, 42(7):992–1003.

Harrison JM, Keskin NB, Zeevi A (2012) Bayesian dynamic pricing policies: Learning and earning under a binary prior distribution. *Management Science*, 58(3):570–586.

Hartigan JA, Wong MA (1979) Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108.

Hazan E, Agarwal A, Kale S (2007) Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192.

Hazan E, et al. (2016) Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325.

Herlocker JL, Konstan JA, Riedl J (2000) Explaining collaborative filtering recommendations. *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work*, 241–250 (ACM).

Honhon D, Gaur V, Seshadri S (2010) Assortment planning and inventory decisions under stockout-based substitution. *Operations Research*, 58(5):1364–1379.

Hopp WJ, Xu X (2008) A static approximation for dynamic demand substitution with applications in a competitive market. *Operations Research*, 56(3):630–645.

Horn RA, Horn RA, Johnson CR (1990) *Matrix analysis* (Cambridge University Press).

Hu K, Acimovic J, Erize F, Thomas DJ, Van Mieghem JA (2018) Forecasting new product life cycle curves: Practical approach and empirical analysis: Finalist–2017 m&som practice-based research competition. *Manufacturing & Service Operations Management*, 21(1):66–85.

Jagabathula S, Rusmevichientong P (2015) A nonparametric joint assortment and price choice model. *Available at SSRN 2286923*.

Jagabathula S, Subramanian L, Venkataraman A (2018) A model-based embedding technique for segmenting customers. *Operations Research*, 66(5):1247–1267.

Javanmard A, Nazerzadeh H (2019) Dynamic pricing in high-dimensions. *Journal of Machine Learning Research*, 20(9):1–49.

Jolliffe I (2011) *Principal Component Analysis* (Springer, Berlin Heidelberg).

Jun KS, Bhargava A, Nowak R, Willett R (2017) Scalable generalized linear bandits: Online computation and hashing. *Advances in Neural Information Processing Systems*, 99–109.

Kaban A (2015) Improved bounds on the dot product under random projection and random sign projection. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 487–496.

Kallus N, Udell M (2016) Dynamic assortment personalization in high dimensions. *arXiv preprint arXiv:1610.05604*.

Kaufmann E, Cappé O, Garivier A (2012) On bayesian upper confidence bounds for bandit problems. *Artificial Intelligence and Statistics*, 592–600.

Keskin NB, Zeevi A (2014) Dynamic pricing with an unknown demand model: Asymptotically optimal semi-myopic policies. *Operations Research*, 62(5):1142–1167.

Keskin NB, Zeevi A (2016) Chasing demand: Learning and earning in a changing environment. *Mathematics of Operations Research*, 42(2):277–307.

Kök AG, Fisher ML, Vaidyanathan R (2015) Assortment planning: Review of literature and industry practice. Agrawal N, Smith SA, eds., *Retail Supply Chain Management*, 175–236 (Springer, Boston).

Kök AG, Xu Y (2011) Optimal and competitive assortments with endogenous pricing under hierarchical consumer choice models. *Management Science*, 57(9):1546–1563.

Lai TL, Robbins H (1985) Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22.

Lattimore T, Szepesvári C (2018) Bandit algorithms. *preprint*.

Lei YM, Jasin S, Sinha A (2014) Near-optimal bisection search for nonparametric dynamic pricing with inventory constraint. *Available at SSRN 2509425*.

Li G, Rusmevichientong P, Topaloglu H (2015) The d-level nested logit model: Assortment and price optimization problems. *Operations Research*, 63(2):325–342.

Li H, Huh WT (2011) Pricing multiple products with the multinomial logit and nested logit models: Concavity and implications. *Manufacturing & Service Operations Management*, 13(4):549–563.

Li L, Chu W, Langford J, Schapire RE (2010) A contextual-bandit approach to personalized news article recommendation. *Proceedings of the 19th International Conference on World Wide Web*, 661–670.

Li L, Chu W, Langford J, Wang X (2011) Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. *Proceedings of the fourth ACM international conference on Web search and data mining*, 297–306 (ACM).

Li L, Lu Y, Zhou D (2017a) Provable optimal algorithms for generalized linear contextual bandits. *arXiv preprint arXiv:1703.00048*.

Li L, Lu Y, Zhou D (2017b) Provably optimal algorithms for generalized linear contextual bandits. *arXiv preprint arXiv:1703.00048*.

Lobel I, Leme RP, Vladu A (2018) Multidimensional binary search for contextual decision-making. *Operations Research*, 66(5):1346–1361.

MacQueen J, et al. (1967) Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, 281–297.

Mahajan S, Van Ryzin G (2001) Stocking retail assortments under dynamic consumer substitution. *Operations Research*, 49(3):334–351.

May BC, Korda N, Lee A, Leslie DS (2012) Optimistic bayesian sampling in contextual-bandit problems. *Journal of Machine Learning Research*, 13(June):2069–2106.

McCullagh P, Nelder JA (1989) *Generalized linear models*, volume 37 (CRC press).

Murtagh F (1983) A survey of recent advances in hierarchical clustering algorithms. *The Computer Journal*, 26(4):354–359.

Nambiar M, Simchi-Levi D, Wang H (2018) Dynamic learning and price optimization with endogeneity effect. *Forthcoming at Management Science*.

Nguyen TT, Lauw HW (2014) Dynamic clustering of contextual multi-armed bandits. *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, 1959–1962 (ACM).

Petrov VV (2012) *Sums of independent random variables*, volume 82 (Springer Science & Business Media).

Qiang S, Bayati M (2016) Dynamic pricing with demand covariates. *Available at SSRN 2765257*.

Rodríguez B, Aydın G (2011) Assortment selection and pricing for configurable products under demand uncertainty. *European Journal of Operational Research*, 210(3):635–646.

Rusmevichientong P, Shen ZJM, Shmoys DB (2010) Dynamic assortment optimization with a multinomial logit choice model and capacity constraint. *Operations Research*, 58(6):1666–1680.

Rusmevichientong P, Shmoys D, Tong C, Topaloglu H (2014) Assortment optimization under the multinomial logit model with random choice parameters. *Production and Operations Management*, 23(11):2023–2039.

Rusmevichientong P, Tsitsiklis JN (2010) Linearly parameterized bandits. *Mathematics of Operations Research*, 35(2):395–411.

Russo D, Van Roy B (2014) Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 39(4):1221–1243.

Ryzin Gv, Mahajan S (1999) On the relationship between inventory costs and variety benefits in retail assortments. *Management Science*, 45(11):1496–1509.

Sarwar BM, Karypis G, Konstan JA, Riedl J, et al. (2001) Item-based collaborative filtering recommendation algorithms. *Www*, 1:285–295.

Sauré D, Zeevi A (2013) Optimal dynamic assortment planning with demand learning. *Manufacturing & Service Operations Management*, 15(3):387–404.

Saxena A, Prasad M, Gupta A, Bharill N, Patel OP, Tiwari A, Er MJ, Ding W, Lin CT (2017) A review of clustering techniques and developments. *Neurocomputing*, 267:664–681.

Sharma A, Hofman JM, Watts DJ (2015) Estimating the causal impact of recommendation systems from observational data. *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, 453–470 (ACM).

Shi J, Malik J (2000) Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905.

Smith BC, Leimkuhler JF, Darrow RM (1992) Yield management at american airlines. *Interfaces*, 22(1):8–31.

Snoek J, Larochelle H, Adams RP (2012) Practical bayesian optimization of machine learning algorithms. *Advances in Neural Information Processing Systems*, 2951–2959.

Strehl A, Langford J, Li L, Kakade SM (2010) Learning from logged implicit exploration data. *Advances in Neural Information Processing Systems*, 2217–2225.

Su Q, Chen L (2015) A method for discovering clusters of e-commerce interest patterns using click-stream data. *Electronic Commerce Research and Applications*, 14(1):1–13.

Talebian M, Boland N, Savelsbergh M (2012) Assortment and pricing with demand learning. *Optimization Online, January*.

Tibshirani R (1996) Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288.

Tran-Thanh L, Chapman A, Cote EMd, Rogers A, Jennings NR (2010) ε-first policies for budget-limited multi-armed bandits. *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, 1211–1216 (AAAI Press).

Tropp JA (2011) User-friendly tail bounds for matrix martingales. ACM Report 2011–01, California Inst. of Tech. Pasadena, CA.

Van der Vaart AW (1998) Asymptotic Statistics (Cambridge University Press, Cambridge).

Van Kampen TJ, Akkerman R, Pieter van Donk D (2012) Sku classification: a literature review and conceptual framework. *International Journal of Operations & Production Management*, 32(7):850–876.

Von Luxburg U (2007) A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416.

Wainwright MJ (2019) *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48 (Cambridge University Press).

Wang R (2012) Capacitated assortment and price optimization under the multinomial logit model. *Operations Research Letters*, 40(6):492–497.

Wang Y, Chen X, Zhou Y (2018) Near-optimal policies for dynamic multinomial logit assortment selection models. *arXiv preprint arXiv:1805.04785*.

Wang Z, Deng S, Ye Y (2014) Close the gaps: A learning-while-doing algorithm for single-product revenue management problems. *Operations Research*, 62(2):318–331.

Yano CA, Gilbert SM (2005) Coordinated pricing and production/procurement decisions: A review. *Managing Business Interfaces*, 65–103.

Zhou L (2015) A survey on contextual multi-armed bandits. *arXiv preprint arXiv:1508.03326*.