A Dynamic Clustering Approach to Data-Driven Assortment Personalization

Fernando Bernstein* • Sajad Modaresi* • Denis Sauré†

* *The Fuqua School of Business, Duke University, Durham, NC 27708-0120*

† *Industrial Engineering Department, University of Chile, Republica 701, Santiago, Chile*

fernando.bernstein@duke.edu • *sajad.modaresi@duke.edu* • *dsaure@dii.uchile.cl*

September 7, 2017

Abstract

We consider an online retailer facing heterogeneous customers with initially unknown product preferences. Customers are characterized by a diverse set of demographic and transactional attributes. The retailer can personalize the customers' assortment offerings based on available profile information to maximize cumulative revenue. To that end, the retailer must estimate customer preferences by observing transaction data. This, however, may require a considerable amount of data and time given the broad range of customer profiles and large number of products available. At the same time, the retailer can aggregate (pool) purchasing information among customers with similar product preferences to expedite the learning process. We propose a *dynamic clustering* policy that estimates customer preferences by adaptively adjusting customer segments (clusters of customers with similar preferences) as more transaction information becomes available. We test the proposed approach with a case study based on a dataset from a large Chilean retailer. The case study suggests that the benefits of the dynamic clustering policy can be substantial and result (on average) in more than 37% additional transactions compared to a *data-intensive* policy that treats customers independently and in more than 27% additional transactions compared to a *linear-utility* policy that assumes that product mean utilities are linear functions of available customer attributes. We support the insights derived from the numerical experiments by analytically characterizing settings in which pooling transaction information is beneficial for the retailer, in a simplified version of the problem. We also show that there are diminishing marginal returns to pooling information from an increasing number of customers.

Keywords: Data-Driven Assortment Planning, Personalization, Dynamic Clustering, Multi-Armed Bandit

1 Introduction

Motivation and Objective. According to a recent study by eMarketer (eMarketer 2014), worldwide business-to-consumer e-commerce sales will reach $2.357 trillion in 2017. With the rapid growth of online sales, many tech entrepreneurs and traditional retailers are finding unprecedented opportunities to both enhance the customer experience and increase revenue. Assortment personalization is one such opportunity. Personalization refers to the practice of offering a marketing or product mix to each customer based on previously collected data (Arora et al. 2008). These offers are tailored to each customer's taste. The benefits of personalization are twofold: on one hand, it results in higher revenue for the retailer due to the increase in sales that results from providing customers with a set of products that more accurately matches their preferences (Arora et al. 2008); on the other hand, it attracts customer attention and fosters customer loyalty and satisfaction (Ansari and Mela 2003).

Personalization in online retailing is well-documented. For example, Amazon.com uses collaborative filtering to personalize recommendations to its users (Linden et al. 2003, Arora et al. 2008). A recent New York Times article reviews a growing number of start-ups that are investing in highly personalized online shopping technology (Wood 2014). These include Stitch Fix, a women's clothing retailer that periodically sends its customers boxes containing five pieces of clothing personalized to each customer's taste (e.g., size, favorite brand and color, budget). Trunk Club is another similar company catering to male customers. Ropazi is a text messaging-base service that specializes in personalized clothing offerings for kids. As noted in the New York Times article, in personalized shopping "the magic comes from data." Online retailers collect an abundance of customer data (e.g., demographic, transactional, etc.). However, given the broad range of customer profiles, collecting a sufficient amount of transaction data on each customer profile may not be possible. This, in turn, limits the retailer's ability to accurately estimate preferences and offer personalized assortments.

The goal of this paper is to explore the efficient use of data for assortment personalization in online retailing. In particular, we study the revenue impact of adaptively pooling transaction information across customers with similar taste.

Model. We consider an online retailer that sells multiple products over a finite selling season. Customers arrive sequentially and the retailer offers each customer an assortment of products. The retailer may face display or capacity constraints that limit the number of products in the offered assortment. The customer then decides whether or not to make a purchase. The retailer's objective is to maximize expected cumulative revenue over the selling season.

Customers are divided into different profiles according to their observable attributes, such as demographic profiles (e.g., gender, age, location) and past transaction information (e.g., purchase history, payment method). This information is exogenous and available upon arrival via the customer's login information or internet cookies. We assume that, from the perspective of the retailer, customers with a common profile are homogeneous with respect to their product preferences (in practice, the definition of profiles reflects the degree of customer information available to the retailer and the level of granularity that allows the retailer to operationalize other marketing decisions).

A central assumption in this work is that the retailer has limited prior information on customers' preferences. Personalizing assortments thus requires the estimation of such preferences which, in turn, requires observing the customers' purchasing decisions (that are themselves affected by the retailer's assortment decisions). This gives rise to an *exploration* (learning preferences) versus *exploitation* (earning revenue) trade-off. As such, we formulate the assortment selection problem as a multi-armed bandit problem (Thompson 1933, Robbins 1985) with multiple simultaneous plays: each product represents an arm and including a product in the offered assortment is equivalent to pulling that arm – see Caro and Gallien (2007).

Off-the-shelf bandit approaches to the assortment problem call for solving an independent instance for each customer profile. However, practical instances of the problem might have scores of different profiles. This implies that arriving to reasonable preference estimates for a given profile might take an unreasonably large amount of time. More importantly, such an approach ignores the possibility that customers with different profiles may share similar product preferences. In this paper, we show that the retailer can exploit these similarities by pooling information across customers with similar purchasing behavior. To that end, we consider the existence of an underlying mapping from profiles to clusters, where a *cluster* is defined as a set of customer profiles with the same product preferences. This mapping is initially *unknown* to the retailer.

We propose a *dynamic clustering* policy under which the retailer estimates both the underlying mapping of profiles to clusters and the preferences of each cluster. Assortment decisions are based on the estimated mapping by adapting decision rules from traditional bandit algorithms. We use a Bayesian semi-parametric framework, called the Dirichlet Process Mixture model, to represent uncertainty about the mapping of profiles to clusters. This model arises as a natural selection given the discrete nature of the mapping and allows us to draw inference without having to predetermine the number of clusters upfront. The dynamic clustering policy approximates the posterior distribution of the mapping of profiles to clusters as well as that of the preference parameters for each cluster by using a Markov Chain Monte Carlo (MCMC) sampling scheme.

Main Contributions. The contributions of this paper are as follows:

We propose a prescriptive approach, called the dynamic clustering policy, for assortment personalization in an online setting. The proposed policy combines existing tools from the Bayesian data analysis literature (for estimating customer preferences through dynamic clustering) and from the machine learning / operations management literature (to prescribe personalized assortment decisions). This approach is motivated by the retailers' interest in identifying segments of customers with similar preferences and in offering personalized assortments to their customers. Unlike most existing work that focuses on *offline* settings (i.e., using historical data), we propose an *online* tool for assortment personalization that can be implemented in real-time. Moreover, the proposed dynamic clustering policy is fairly general and flexible as the number of clusters (segments) is endogenous and does not need to be pre-determined.

We illustrate the practical value of the dynamic clustering policy in a realistic setting by using a dataset from a large Chilean retailer. We use the case study to quantify the efficiency and demonstrate the implementability of the dynamic clustering policy. The dataset consists of roughly 95,000 customer-tied click records in the retailer's website for 19 products in the footwear category. We contrast the performance of the proposed policy to that of a *data-intensive* policy that ignores any potential similarity in preferences across profiles and thus estimates product preferences for each profile separately. We find that, in the case study, the proposed policy generates more than 37% additional transactions (on average) compared to the data-intensive policy. This finding quantifies the potential benefit of leveraging similarity in customer preferences by adaptively pooling information across profiles with similar purchase behavior to expedite the learning process. The performance of the dynamic clustering policy is remarkable, considering that the underlying customer population in the case study is rather heterogeneous. We also compare the performance of the proposed policy to that of a *linear-utility* policy that assumes a more structured model of customer preferences. In particular, the linear-utility policy assumes that demand is driven by a customer choice model in which mean utilities are linear functions of customer attributes. The findings from the case study show that the proposed policy generates more than 27% additional transactions (on average) compared to the linear-utility policy. While this can be partially explained by the fact that preferences in the dataset do not exhibit a linear structure, the advantage of the dynamic clustering policy persists even when using synthetic data (based on the dataset) under which mean utilities are linear functions of customer attributes by construction. This finding reinforces the benefits of pooling information through the proposed dynamic clustering approach.

To support the insights derived from the numerical experiments, we analyze a simplified version of the dynamic assortment selection problem in which a single product is offered to each arriving customer. First, we compare the performance of the data-intensive policy to that of a *semi-oracle* policy that knows upfront the mapping of customer profiles to clusters and thus conducts preference estimation and assortment optimization independently for each cluster (as opposed to each profile). This policy exploits the structure of preferences across profiles. Aligned with intuition, we show that the semi-oracle policy outperforms the data-intensive policy, indicating that pooling information is beneficial for the retailer. We also show that there are diminishing marginal returns to pooling information from an increasing number of customer profiles. Next, we contrast the performance of the data-intensive policy with that of a pooling policy that aggregates information across all profiles (regardless of whether their preferences are similar or not). This scenario favors the data-intensive policy, as pooling information across all customers may lead to erroneous estimates and thus to suboptimal assortment offerings. Despite its shortcomings, we characterize conditions under which the pooling policy outperforms the data-intensive policy. The result highlights the benefit of pooling information in the short-term, when there is insufficient data to accurately estimate preferences for each customer profile.

Organization of the Paper. Section 2 provides a review of the relevant literature. Section 3 describes the model. Section 4 presents the dynamic clustering policy, while Section 5 illustrates its effectiveness through a case study. Section 6 discusses theoretical results characterizing settings in which pooling information is beneficial for the retailer. Section 7 provides concluding remarks. All proofs are relegated to Appendix A. Appendix B discusses the extension of the results of Section 6 for Thompson Sampling.

2 Literature Review

This paper proposes a prescriptive approach that integrates dynamic clustering and demand learning with assortment personalization. The paper contributes to and lies at the intersection of three streams of research: dynamic assortment planning with demand learning, personalization, and segmentation methods. The literature on dynamic assortment planning with demand learning mostly focuses on a homogeneous population of customers with unknown demand. The second stream of work is generally concerned with assortment or pricing personalization for a heterogeneous population of customers, but does not consider online demand learning or clustering. Finally, from a methodological standpoint, this paper is related to the literature on Bayesian hierarchical and segmentation methods. Most of papers in this stream of work focus on offline settings, given historical data for estimation purposes.

We next review these streams of work in more detail.

Dynamic Assortment Planning with Demand Learning. Our paper contributes to the vast and growing literature on assortment planning. We refer to Kök et al. (2015) for a comprehensive review of the assortment planning literature and industry practices. To the best of our knowledge, there are only a handful of papers that study dynamic assortment planning with demand learning. Using a Bayesian approach to learn customer preferences, Caro and Gallien (2007) formulate the problem as a multi-armed bandit with multiple plays per period and employ a Lagrangian relaxation approach to propose an index policy for dynamic assortment selection. Ulu et al. (2012) study the dynamic assortment decisions of a firm with horizontally differentiated products for which customers have heterogeneous tastes modeled as locations on a Hotelling line. Following a frequentist approach to estimate customer preferences, Rusmevichientong et al. (2010) study an assortment optimization problem with capacity constraints under the Multinomial Logit (MNL) model. Sauré and Zeevi (2013) study a similar problem under a more general random utility model which subsumes the MNL as a special case. They prove a fundamental limit on the achievable performance of any admissible policy. Using this bound, they propose adaptive policies that balance the implied exploration versus exploitation trade-off. We use elements of this policy in the numerical study in Section 5. In a similar setting, Agrawal et al. (2016) propose an online upper-confidence-bound-type policy for which they prove a performance upper bound. Most of the papers in this stream of work assume a homogeneous population of customers and therefore do not focus on clustering or assortment personalization.

Our work is also related to the literature on dynamic pricing with demand learning. Ferreira et al. (2016) consider a price-based network revenue management problem and extend the Thompson Sampling policy to a setting that involves inventory constraints. Cheung et al. (2016) study a dynamic pricing problem where the demand function is unknown but belongs to a known finite set. They propose an online policy and prove performance bounds. We refer the reader to Besbes and Zeevi (2009), Harrison et al. (2012), and the references therein.

Personalization. Personalization has been studied in the marketing literature from both conceptual and methodological perspectives. Murthi and Sarkar (2003) present a review of research on personalization with a focus on learning customer preferences. See also Montgomery and Smith (2009) and Arora et al. (2008) for two more recent reviews of past research on personalization and some of its examples in practice. In the operations management literature, Bernstein et al. (2015) study a dynamic assortment planning problem with limited inventory under a mixed Logit choice model. The authors prove structural properties of the optimal policy in certain settings and propose a heuristic for assortment customization. In a similar setting, Golrezaei et al. (2014) study an assortment planning problem with a general choice model and limited inventories. They propose an index-based policy and prove performance bounds. Chen et al. (2015) consider personalized pricing (or assortment) decisions in an offline setting given customers' contextual information, using Logit models. Gallego et al. (2015) study a problem of resource allocation with applications to personalized assortment optimization. Jasin and Kumar (2012) propose certainty-equivalent heuristics for a network revenue management problem with customer choice where the seller chooses a collection of products to offer to customers based on their type. Ciocan and Farias (2014) propose a demand estimation algorithm for a high-dimensional network revenue management problem for online display advertising. Kallus and Udell (2016) study a high-dimensional dynamic assortment personalization problem where the number of customer types is large. Assuming that the underlying parameter matrix is of low rank, they use a nuclear-norm regularized maximum likelihood approach for estimation. While these papers study assortment personalization, they do not integrate assortment decisions with online demand learning or clustering.

Segmentation Methods and Methodological Background. Customer segmentation has been widely studied in the marketing literature. Wedel and Kamakura (2012) provide a comprehensive review of market segmentation methodologies such as clustering, mixture models, and profiling segments. In our paper, the proposed Bayesian representation of uncertainty on the mapping of profiles to clusters is based on the Dirichlet Process Mixture model – see Heller and Ghahramani (2005) for a Bayesian hierarchical clustering algorithm which can be used as a deterministic alternative (approximation) to MCMC inference in Dirichlet Process Mixture models. Our adaptation of the Dirichlet Process Mixture model is based on Neal (2000). While, to the best of our knowledge, ours is the first paper in the operations management literature to employ this model, researchers in other fields have used the Dirichlet Process Mixture model to capture heterogeneity in the customers' population. In the marketing literature, Ansari and Mela (2003) use this model to customize the design and content of emails to increase web traffic (click-through). In the economics literature, Burda et al. (2008) use this model to estimate the parameters of a Logit-Probit choice model. In these papers, the estimation is conducted offline using historical data.

As noted earlier, the dynamic assortment selection problem we study can be thought of as a multi-armed bandit problem with multiple plays per period. In their seminal work, Lai and Robbins (1985) prove a fundamental limit on the achievable performance of any so-called consistent policy in a classic bandit setting (we use this lower bound for the analysis in Section 6). Anantharam et al.

(1987) extend the fundamental limit of Lai and Robbins (1985) to a multi-armed bandit problem with multiple plays. The dynamic assortment selection problem can alternatively be formulated as a combinatorial multi-armed bandit problem envisioning each feasible assortment as an arm. This is the approach used in Rusmevichientong et al. (2010) and Sauré and Zeevi (2013). We refer the reader to Modaresi et al. (2014) and the references therein for a combinatorial multi-armed bandit formulation and a review of the relevant literature.

3 Model and Preliminaries

In this section, we formalize the retailer's assortment personalization problem. In particular, we introduce the notion of *clusters*, which captures the presence of heterogeneity in the customer population, and discuss the connection of the model to the multi-armed bandit problem.

Problem Definition. Consider an online retailer endowed with N products and let T denote the total number of customers that arrive during the selling season. Let $\mathcal{N} := \{1, \ldots, N\}$ denote the set of all products. The retailer has a limited display or capacity constraint of C products, i.e., the retailer can show a selection of at most C products to each arriving customer. Without loss of generality, we assume that $C \leq N$. Such display constraints have been motivated and used in different settings in previous studies (see, e.g., Besbes and Sauré (2016), Fisher and Vaidyanathan (2014), Rusmevichientong et al. (2010), and Caro and Gallien (2007)). In an online retail setting, this constraint may be related to limitations on the time customers spend searching for a product, or the number of products displayed in the webpage. In the context of companies such as Ropazi (mentioned in Section 1), customers are shown a subset of the entire product set. For $j \in \mathcal{N}$, we let r_j denote product j's unit price, which we assume to be fixed throughout the selling season.

Customers arrive sequentially throughout the selling season. We use t to index customers according to their arrival times, so that time $t = 1$ corresponds to the first customer arrival and time $t = T$ to the last one. The retailer classifies customers according to their profile information. Each profile is encoded as a unique vector of attributes (e.g., gender, age, location, past transactions, payment method). For example, in the case study presented in Section 5, we use customers' gender, age, and location to define their profile information. Thus, in that case, a customer profile is described by the attribute vector $x = (x_{gender}, x_{age}, x_{location})$. The profile of an arriving customer is observed by the retailer via the customer's login information or internet cookies. Let $\mathcal{I} := \{1, \ldots, I\}$ denote the set of customer profiles, where each profile i is associated with a unique vector of attributes x^i, $i \in \mathcal{I}$. A customer with profile i arrives with probability p_i, where $0 < p_i < 1$ for $i \in \mathcal{I}$ and $\sum_{i \in \mathcal{I}} p_i = 1$. Let $i_t \in \mathcal{I}$ denote the profile of customer t. Upon arrival, a customer is offered an assortment of at most C products. Let $\mathcal{S}$ denote the set of feasible assortments, i.e., $\mathcal{S} := \{S \subseteq \mathcal{N} : |S| \leq C\}$, where $|S|$ denotes the cardinality of set S, and let $S_t \in \mathcal{S}$ denote the assortment offered to customer t.

Demand Model. The retailer's revenue is contingent on the customers' purchasing decisions. Let $Z^i_{j,t}$ denote the purchasing decision of a customer with profile i arriving at time t regarding product $j \in S_t$. More specifically, $Z^i_{j,t} = 1$ if customer t is from profile i and purchases product $j \in S_t$ and $Z^i_{j,t} = 0$, otherwise. We consider two cases in terms of the underlying demand model:

MNL Demand. In this setting, a customer with profile i arriving at time t assigns a (random) utility $U^i_{j,t}$ to product j, with

$$U^i_{j,t} := \mu^i_j + \zeta^i_{j,t}, \quad j \in \mathcal{N} \cup \{0\},$$

where μ^i_j is the mean utility of product j for profile i (which is unknown to the retailer), $\zeta^i_{j,t}$ are independent (across i, j, and t) and identically distributed random variables drawn from a standard Gumbel distribution, and product 0 denotes the no-purchase alternative. We assume, without loss of generality, that $\mu^i_0 = 0$ for all $i \in \mathcal{I}$. We do not assume any particular relation between the mean utilities and customer attributes. In Section 5.5, we compare this model to another approach that assumes that mean utilities are linear functions of customer attributes. When offered an assortment $S_t \in \mathcal{S}$, customer t selects the product with the highest utility (including, possibly, the no-purchase alternative).

For a given assortment $S \in \mathcal{S}$ and a given vector of mean utilities $\mu^i := (\mu^i_1, \ldots, \mu^i_N)$, let $\Pi_j(S, \mu^i)$ denote the probability that a customer with profile i purchases product $j \in S \cup \{0\}$. We have that

$$\Pi_j(S, \mu^i) := \frac{\nu^i_j}{1 + \sum_{j' \in S} \nu^i_{j'}}, \; j \in S \cup \{0\}, \quad \Pi_j(S, \mu^i) := 0, \text{ otherwise}, \tag{1}$$

where $\nu^i_j := \exp(\mu^i_j)$ are the exponentiated mean utilities for $j \in \mathcal{N} \cup \{0\}$. Thus, for a customer with profile i arriving at time t and offered assortment S_t, $Z^i_{j,t} = 1$ with probability $\Pi_j(S_t, \mu^i)$. Moreover, we let $Z^i_{0,t} = 1$ if a customer with profile i arriving at time t opts not to purchase any product, and $Z^i_{0,t} = 0$ otherwise.

Independent Demand. In this setting, we assume that the purchasing decision $Z^i_{j,t}$ of a customer with profile i arriving at time t for product j is a Bernoulli random variable independent of the customer's purchasing decision for the other products[1]. In particular, we assume that a customer with profile i purchases product $j \in S_t$ with probability μ^i_j, independent of the assortment in which it is offered. Thus, in this setting, the vector $\mu^i = (\mu^i_1, \ldots, \mu^i_N)$ represents the purchase probabilities (i.e., mean of the Bernoulli distributions) for a customer with profile i. These purchase probabilities are unknown to the retailer. For a given assortment $S \in \mathcal{S}$ and a given vector of purchase probabilities μ^i, we let $\Pi_j(S, \mu^i)$ denote the probability that a customer with profile i purchases product $j \in S$, i.e.,

$$\Pi_j(S, \mu^i) := \mu^i_j, \quad j \in S. \tag{2}$$

A setting in which this demand model may be appropriate is one in which customers make purchas-

[1] We focus on the Bernoulli distribution for clarity of exposition. The framework introduced in this paper applies to other distributions as well.

ing decisions across various products that are not necessarily substitutes, as may be in the cases of Stitch Fix and Ropazi mentioned in Section 1.

Because of the dual role of the parameters $(\mu^i, i \in \mathcal{I})$ in these two demand models (as mean utilities under MNL demand and as purchase probabilities under independent demand), throughout the paper, we use the terms *purchase probabilities* and *preferences* interchangeably.

Assortment Selection. Let $W_t := (i_t, Z_t)$ denote the profile of customer t together with the vector of purchasing decisions, where $Z_t := (Z^{i_t}_{1,t}, \ldots, Z^{i_t}_{N,t})$. Let $\mathcal{F}_t := \sigma\left((S_\tau, W_\tau), 1 \leq \tau \leq t\right), t = 1 \ldots, T$, denote the filtration (history) associated with the assortment and purchasing decisions up to (and including) time t, with $\mathcal{F}_0 = \emptyset$. An admissible assortment selection policy π is a mapping from the available history to assortment decisions such that $S_t \in \mathcal{S}$ is non-anticipating (i.e., S_t is $\mathcal{F}_{t-1}$-measurable) for all t. Let $\mathcal{P}$ denote the set of admissible policies. The retailer's objective is to choose an assortment selection policy $\pi \in \mathcal{P}$ to maximize expected cumulative revenue over the selling season:

$$J^\pi(T, I) := \mathbb{E}_\pi \left(\sum_{t=1}^{T} \sum_{i=1}^{I} \sum_{j \in S_t} r_j Z^i_{j,t} \right),$$

where $\mathbb{E}_\pi$ denotes the expectation when policy $\pi \in \mathcal{P}$ is used.

Market Heterogeneity (Clusters). Although profiles differ in their observable attributes, customers with different profiles may have similar preferences for products. We define a *cluster* (or *segment*) as a set of customer profiles that have identically distributed preferences.[2] (We use the terms cluster and segment interchangeably in the paper.) This implies the existence of an underlying mapping of profiles to clusters $M : \mathcal{I} \to \mathcal{K}$, where $\mathcal{K} := \{1, \ldots, K\}$ is the set of cluster labels and $K \leq I$ is the number of clusters. The mapping M assigns a cluster label $M(i) \in \mathcal{K}$ to each profile $i \in \mathcal{I}$, so that any two profiles with the same cluster label share the same set of preference parameters. That is, $\mu^i = \mu^{i'}$ if i and i' are such that $M(i) = M(i')$. The underlying mapping M of profiles to clusters is unknown to the retailer. In this regard, the case study presented in Section 5 as well as the analytical results in Section 6 show that the retailer benefits from estimating the mapping of profiles to clusters as it helps to expedite the estimation of preferences. This, in turn, translates into higher revenue for the retailer.

Connection to the Multi-Armed Bandit Problem. The retailer does not know the customers' preferences, so assortment personalization requires estimating such preferences by observing the customers' purchasing decisions. The history of purchasing decisions is, in turn, affected by past assortment decisions. This leads to an *exploration* (learning preferences) versus *exploitation* (earning revenue) trade-off. The multi-armed bandit problem is the standard framework for addressing this trade-off. The assortment selection problem can be formulated as a multi-armed bandit by means

[2] In order to formalize the definition of a cluster, we define it as a set of customer profiles with identically distributed preferences. However, the notion of "similar taste" is embedded in the dynamic clustering algorithm (introduced in Section 4) that produces the clusters.

of the following analogy: each product $j \in \mathcal{N}$ corresponds to an arm, and offering a product (i.e., including that product in the offered assortment) is equivalent to pulling that arm (see, e.g., Caro and Gallien (2007)). Thus, one can think of the problem as a finite horizon multi-armed bandit with multiple plays per period, where at each point in time, at most C out of N arms are pulled. Following the bandit literature, we restate the retailer's objective of maximizing expected cumulative revenue in terms of the *regret*. To that end, we first define $S_i^* \in \mathcal{S}$ as the optimal assortment that the retailer would offer to customers with profile i if μ^i was known. That is,

$$S_i^* \in \operatorname*{argmax}_{S \in \mathcal{S}} \sum_{j \in S} r_j \, \Pi_j(S, \mu^i).$$

We define the regret associated with any policy π as[3]

$$R^\pi(T, I) := \sum_{i \in \mathcal{I}} p_i \left(\sum_{j \in S_i^*} r_j \, \Pi_j(S_i^*, \mu^i) \right) T - J^\pi(T, I). \tag{3}$$

The regret measures the retailer's expected cumulative revenue loss relative to a clairvoyant retailer that knows the purchase probabilities (and thus the underlying mapping of profiles to clusters). That is, the regret represents the retailer's expected cumulative revenue loss due to the lack of prior knowledge of purchase probabilities which results in suboptimal assortment offerings. Maximizing expected cumulative revenue is equivalent to minimizing the regret over the selling season.

An assortment selection policy in this setting is comprised of two elements: an *estimation* tool for estimating the customers' preferences and an *optimization* tool for deciding what assortment to offer to each arriving customer. As stated earlier, we focus on bandit algorithms as the optimization tool (we discuss this in more detail in Section 4.4). When comparing the performance of different policies, we assume that they follow the same bandit algorithm.

Model Discussion. This paper focuses on the efficient use of information to make personalized assortment offerings. In particular, we investigate the retailer's potential revenue benefit from aggregating transaction information across customers with similar product preferences. To this end, we make a few assumptions to facilitate the study. We assume perfect inventory replenishment for the retailer, and that the retailer incurs no operational costs (e.g., switching costs) for offering different assortments to different customers. Such assumptions are common in the dynamic assortment planning literature and allow us to isolate the role of dynamic personalized assortment planning in maximizing retailer's revenue. We also assume that the products' prices are constant throughout the selling season. This assumption is also common in the assortment planning literature and facilitates analysis (see, e.g., Sauré and Zeevi (2013)). Finally, we assume that customers' purchasing decisions are independent over time and across customers (i.e., we ignore word-of-mouth and other related effects).

[3] Note that $J^\pi(T, I)$ and $R^\pi(T, I)$ are functions of μ^i and p_i for all $i \in \mathcal{I}$ as well, but we drop such dependence to simplify the notation.

4 Dynamic Assortment Personalization

In this section, we introduce a prescriptive approach for dynamic assortment personalization which we call the *dynamic clustering* policy. This policy adaptively estimates both the customers' preferences and the mapping of profiles to clusters in a Bayesian manner. In what follows, we first present the Bayesian model of preferences in Section 4.1, followed by the dynamic clustering policy in Section 4.2. Section 4.3 discusses the estimation procedure based on the observed purchase history, while Section 4.4 reviews the bandit policies we use. We illustrate the performance of the dynamic clustering policy in a case study in Section 5.

4.1 Bayesian Model of Preferences

In this section, we present a Bayesian framework to model customers' preferences. In section 4.3, we present a Markov Chain Monte Carlo (MCMC) sampling technique to estimate the model discussed in this section.

Recall that $Z^i_{j,t}$ denotes the random variable that captures the purchasing decision of a customer with profile i arriving at time t regarding product $j \in S_t$. We define $Z^i_t := (Z^i_{1,t}, \ldots, Z^i_{N,t})$ and let $F(\cdot|\mu^i)$ denote the distribution of Z^i_t as a function of the vector of parameters μ^i. This distribution is independent of t as preferences are time-homogeneous. For the case of MNL demand, $F(Z^i_t = e^j|\mu^i) = \Pi_j(S, \mu^i)$, where $\Pi_j(S, \mu^i)$ is as defined in (1) and e^j denotes the j-th unit vector (although $F(\cdot|\mu^i)$ also depends on the assortment S, we drop such dependence to simplify notation). For the case of independent demand, $F(Z^i_{j,t} = 1|\mu^i) = \Pi_j(S, \mu^i)$, where $\Pi_j(S, \mu^i)$ is as defined in (2) and $Z^i_{j,t}$ are independent across j. The retailer knows the family of distributions F, but does not know the vector of parameters μ^i that characterizes this distribution for customers with profile i.

We adopt a hierarchical Bayesian model to represent the retailer's uncertainty with respect to the underlying mapping M from profiles to clusters and to the vector of parameters μ^i governing the preferences of customer profiles. More specifically, we model the distribution from which the Z^i_t's are drawn as a mixture of distributions of the form $F(\cdot|\mu^i)$. We denote the mixing distribution over μ^i by H and let the prior distribution of H be a Dirichlet Process (Ferguson 1973, Antoniak 1974). A Dirichlet Process prior is a natural selection as its realizations are (discrete) probability distributions. The Dirichlet Process is specified by a distribution H_0, which serves as a baseline prior for H, and a precision parameter α (which is a positive real number) that modulates the deviations of H from H_0 – the larger the precision parameter α, the more concentrated the Dirichlet Process prior is around the baseline location H_0. We denote the Dirichlet Process by $\mathrm{DP}(H_0, \alpha)$. We therefore model the uncertainty over customers' preferences as follows:

$$Z^i_t|\mu^i \sim F(\cdot|\mu^i) \tag{4a}$$

$$\mu^i|H \sim H \tag{4b}$$

$$H \sim \mathrm{DP}(H_0, \alpha). \tag{4c}$$

Being an infinite mixture model, the Dirichlet Process Mixture provides a flexible framework for capturing heterogeneity in the customer population without the need to predetermine the number of clusters. In fact, the number of clusters is endogenously determined based on the observed transaction data. Further details on the Dirichlet Process can be found in Ferguson (1973) and Ferguson (1983).

4.2 Dynamic Clustering Policy

We introduce the dynamic clustering policy by presenting a general description of the sequence of events that takes place for each customer arrival t. Let Φ denote the set of time indices (periods) in which the dynamic clustering policy updates the mapping of profiles to clusters. For instance, in the case study in Section 5, we take $\Phi = \{100, 200, 300, \ldots\}$; that is, we update the mapping after every 100 customer arrivals (to expedite the computation time of the algorithm).

Step 1 (Arrival). Observe the profile of the arriving customer t (i.e., i_t).

Step 2 (Assortment Selection). Follow a bandit algorithm to determine the assortment $S_t \in \mathcal{S}$ to offer customer t based on the current preference and mapping estimates. (For the first customer arrival, start with an arbitrary mapping that randomly assigns profiles to clusters and select preferences randomly.) See Section 4.4 for details.

Step 3 (Transaction). Observe the purchasing decision of customer t and update the assortment and purchase history.

Step 4 (Mapping and Preference Estimation). If $t \in \Phi$, then perform the estimation procedure described in Section 4.3 to approximate the posterior distribution of the parameters in model (4) given the updated history; otherwise, only update the preference estimates using the prevailing mapping estimate.

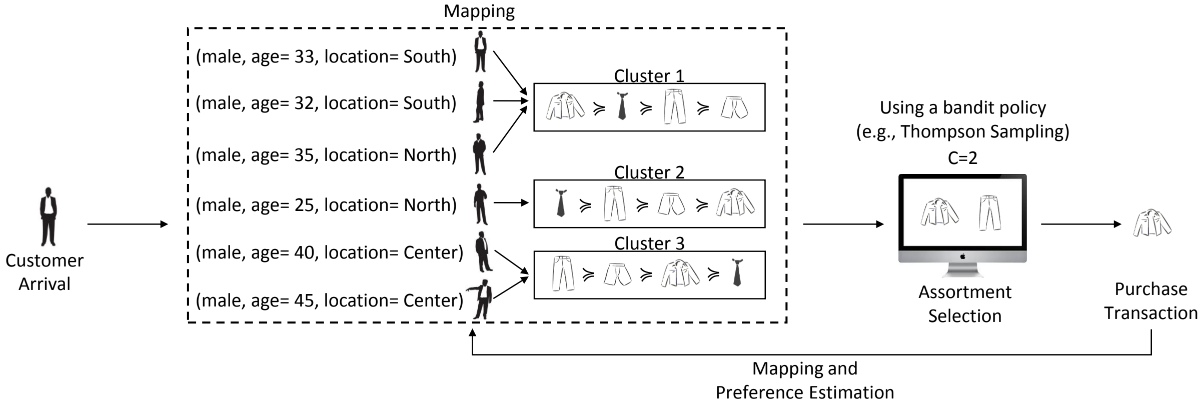
Mapping
(male, age= 33, location= South)
(male, age= 32, location= South)
(male, age= 35, location= North)
(male, age= 25, location= North)
(male, age= 40, location= Center)
(male, age= 45, location= Center)
Cluster 1
Cluster 2
Cluster 3
Customer Arrival
Using a bandit policy (e.g., Thompson Sampling)
C=2
Assortment Selection
Purchase Transaction
Mapping and Preference Estimation

Figure 1: Illustration of the dynamic clustering policy.

Figure 1 illustrates the steps in the dynamic clustering policy. This policy adapts existing tools from the Bayesian data analysis and machine learning/operations management literature. In Step 2 (i.e., optimization step), we use a bandit algorithm to determine the assortment to offer each arriving customer based on the current mapping and preference estimates. We discuss the implementation details of the bandit algorithm in Section 4.4. In Step 4 (i.e., estimation step), we implement the estimation procedure that will be introduced in Section 4.3 to update the mapping of profiles to clusters and the corresponding preference estimates associated with each cluster.

4.3 Mapping and Preference Estimation

We estimate the mapping and preferences in Step 4 of the dynamic clustering policy by approximating the posterior distribution of the parameters in model (4). To that end, we use an MCMC sampling scheme comprised of a Metropolis-Hastings step to update the mapping of profiles to clusters followed by a Gibbs sampling step to update the posterior distribution of preference parameters for each cluster. This implementation is an adaptation of the sampling scheme in Neal (2000), which is tailored for the case of a Dirichlet Process Mixture model. The output of the MCMC sampling scheme is a sequence of mappings of profiles to clusters and preference parameter vectors. The sampling procedure is tailored so that the samples approximate a set of independent draws from the posterior distribution of the model parameters. (See Gelman et al. (2014) for further details on MCMC methods.)

Next, we provide details on the sampling scheme within the MCMC procedure. Consider an arriving customer t, with $t \in \Phi$. After observing the profile of the customer, offering an assortment, and recording the purchasing decision, the dynamic clustering algorithm approximates the posterior distribution of the model parameters (mapping and preferences). Let $\mathcal{X}_t^i$ denote the assortment and purchase history associated with profile i up to (and including) time $t \leq T$. That is, $\mathcal{X}_t^i := \left\{(S_u, Z_u^{i_u}) : i_u = i, 1 \leq u \leq t\right\}$, and set $\mathcal{X}_0^i = \emptyset$. Define $\mathcal{X}_t := (\mathcal{X}_t^1, \cdots, \mathcal{X}_t^I)$. The following sampling procedure generates a sequence of mappings and parameter vectors (one for each cluster in each mapping). Let the tuning parameter η denote the number of samples to be collected.

MCMC Sampling. Start with an arbitrary mapping M_1 that assigns profiles to clusters and sample the preference parameters from H_0 for each cluster. For $s = 1, \ldots, \eta$, repeat the sampling process as follows:

- **Step 1 (Cluster Update).** Let $c_i = M_s(i)$ denote the cluster associated with profile i under the mapping M_s and let c denote a generic cluster. For each profile $i \in \mathcal{I}$, update the cluster label c_i associated with that customer profile as follows. Let $n_{-i,c}$ be the number of profiles, excluding profile i, that are mapped to an existing cluster c under the mapping M_s. Draw a candidate cluster label c_i^* according to the following probability distribution:[4]

$$\begin{aligned}\mathbb{P}\,(\text{assign } c_i \text{ to an existing cluster } c) &= \frac{n_{-i,c}}{I-1+\alpha} \\ \mathbb{P}\,(\text{assign } c_i \text{ to a new cluster}) &= \frac{\alpha}{I-1+\alpha}.\end{aligned}$$

If $c_i^* \in \{c_1, \dots, c_I\}$, then use the corresponding parameter vector $\mu^{c_i^*}$. If $c_i^* \notin \{c_1, \dots, c_I\}$, i.e., if the candidate cluster does not correspond to any of the existing clusters under M_s, then sample $\mu^{c_i^*}$ from H_0.

Set the new value of c_i to c_i^* with probability

$$a(c_i^*, c_i) := \min\left\{1, \frac{L(\mathcal{X}_t^i, \mu^{c_i^*})}{L(\mathcal{X}_t^i, \mu^{c_i})}\right\}$$

and do not change c_i with probability $1 - a(c_i^*, c_i)$, where $L(\mathcal{X}_t^i, \mu^{c_i})$ denotes the likelihood function given the purchase history $\mathcal{X}_t^i$ and the vector of parameters μ^{c_i}. Let M_{s+1} be the updated mapping given by the new assignment of profiles to clusters (i.e., updated c_i's).

- **Step 2 (Preference Update).** Update the vector of preference parameters for each cluster: for each $c \in \{c_1, \dots, c_I\}$, compute the posterior distribution of μ^c (given the history $\mathcal{X}_t$) and draw a new realization for μ^c from its posterior distribution.

To approximate the posterior distribution of the mapping, we discard the first η_b samples drawn ("burn-in" period), and select every other η_d-th draw (e.g., every 10th draw) from the remaining samples (both η_b and η_d are tuning parameters). Let $m' = (\eta - \eta_b)/\eta_d$ denote the number of MCMC draws used for estimation. Denote the corresponding (distinct) mappings by $M_1, M_2, \dots, M_m$, with $m \leq m'$ as the mappings corresponding to several sample points may be identical. Let $0 < f_1, f_2, \dots, f_m \leq 1$ be the associated frequency proportions (i.e., the relative number of occurrences of each mapping in the set of selected samples). We approximate the posterior distribution of the mapping as a discrete probability distribution that takes the value M_n with probability f_n, $n \leq m$. Note that the number of possible mappings from profiles to clusters is combinatorial in I, the number of profiles. In this regard, the approximation we propose alleviates the complexity of calculating the posterior distribution of the mapping. We discuss further implementation details in Section 5.

The preference update in Step 2 above depends on the underlying demand model. Under MNL demand, it is necessary to introduce a separate Metropolis-Hastings step to update the posterior distribution of μ^c, as there is no conjugate prior for the MNL model. This, however, comes at the expense of additional computational effort. To alleviate this computational burden, we approximate the parameters of the MNL model by using frequentist point estimates. (Because of this approximation, we do not need to specify the prior distribution H_0.) Specifically, suppose that

[4] This distribution is derived in Neal (2000), where α is the precision parameter of the Dirichlet Process Mixture model.

$t-1$ customers have arrived so far and that the first step of the MCMC has resulted in a mapping of profiles to clusters denoted by M_s. In Step 2 of the MCMC, we estimate the exponentiated mean utility ν_j^i by $\hat{\nu}_{j,t}^{M_s(i)}$, where

$$\hat{\nu}_{j,t}^{M_s(i)} := \frac{\sum_{l=1}^{t-1} Z_{j,l}^{i_l} \, \mathbf{1}\left\{j \in S_l, M_s(i_l) = M_s(i)\right\}}{\sum_{l=1}^{t-1} Z_{0,l}^{i_l} \, \mathbf{1}\left\{j \in S_l, M_s(i_l) = M_s(i)\right\}}, \; j \in \mathcal{N}. \tag{6}$$

We then estimate $\mu_j^{M_s(i)}$ by $\hat{\mu}_{j,t}^{M_s(i)} := \ln(\hat{\nu}_{j,t}^{M_s(i)})$. Note that this parameter estimation is conducted at the product level by exploiting the independence of irrelevant alternatives (IIA) property of the MNL model. Moreover, such estimates are obtained for each cluster by pooling transaction data across customers within the same cluster. The numerical results reported in Section 5 suggest that this approximation results in a reasonable performance and computation time. Under independent demand, we take H_0 in the Dirichlet Process Mixture to be the product of independent Beta distributions, as the Beta distribution is the conjugate prior of the Bernoulli distribution. Thus, the posterior distribution of μ^c in Step 2 can be computed in closed-form using Bayes' rule.

4.4 Assortment Optimization

We next describe the bandit policies used for the assortment selection rule.

MNL demand. For the MNL model, we adapt Algorithm 3 of Sauré and Zeevi (2013) to our setting. This algorithm determines whether to explore or exploit for each arriving customer t, as follows. If all products have been explored at least a number of times (which is of order $\ln(t)$), then the algorithm exploits the current optimal assortment. Otherwise, it offers an assortment containing under-tested products (exploration). We refer to Sauré and Zeevi (2013) for further details. Sauré and Zeevi (2013) assume a homogeneous population of customers. However, in the Bayesian setup of the dynamic clustering policy, estimates are derived from the approximation to the posterior distribution of the mapping and customer preferences. Thus, we adapt the algorithm in Sauré and Zeevi (2013) for the dynamic clustering policy. To that end, suppose that $t-1$ customers have arrived so far, and the Mapping and Preference Estimation procedure (discussed in Section 4.3) has resulted in the distinct mappings $M_1, M_2, \ldots, M_m$ with frequency proportions $f_1, f_2, \ldots, f_m$, respectively. Let customer t have profile i. We estimate the exponentiated mean utilities ν_j^i by $\hat{\nu}_{j,t}^i$, where

$$\hat{\nu}_{j,t}^i := \sum_{l=1}^{m} f_l \, \hat{\nu}_{j,t}^{M_l(i)},$$

and $\hat{\nu}_{j,t}^{M_l(i)}$ is as defined in (6). Moreover, we let

$$T_j^i(t) := \sum_{l=1}^{m} f_l \, T_j^{M_l(i)}(t),$$

where $T_j^{M_l(i)}(t)$ is the number of times that product j has been offered to a customer from cluster

$M_l(i)$ up to (and excluding) time t. In other words, $T_j^i(t)$ is the *average* number of times (over different mappings) that product j has been offered to a customer from the cluster associated with profile i up to (and excluding) time t. The quantities $\hat{\nu}_{j,t}^i$ and $T_j^i(t)$ are used to select the assortment to offer customer t in Algorithm 3 of Sauré and Zeevi (2013).

Independent demand. For the independent demand model, we adapt the Thompson Sampling policy (Thompson 1933) to our setting. We first present details of this policy for a classic setting (i.e., a homogeneous population of customers, a single product offering to each customer, and equal product prices) and then discuss how we adapt this policy to our setting.

- In a classic bandit setting, let $Beta(1,1)$, i.e., a uniform distribution, be the conjugate prior of the purchase probability of customers for each product. Let $Beta(a_{j,t}, b_{j,t})$ denote the posterior distribution with parameters $a_{j,t}$ and $b_{j,t}$ for product $j \in \mathcal{N}$, where $a_{j,0} = b_{j,0} = 1$ for all products $j \in \mathcal{N}$, $a_{j,t} = a_{j,t-1} + 1$ and $b_{j,t} = b_{j,t-1}$ if customer t purchases product $j \in S_t$, and $a_{j,t} = a_{j,t-1}$ and $b_{j,t} = b_{j,t-1} + 1$ if customer t does not purchase product $j \in S_t$. Moreover, $a_{j,t} = a_{j,t-1}$ and $b_{j,t} = b_{j,t-1}$ for all other products $j \in \mathcal{N} \setminus S_t$ (i.e., the products that are not offered to customer t). Sample $Q_{j,t}$ randomly from the posterior distribution $Beta(a_{j,t-1}, b_{j,t-1})$, and offer product $S_t \in \underset{j \in \mathcal{N}}{\operatorname{argmax}} \{Q_{j,t}\}$ at time t.
- In our setting, suppose that the Mapping and Preference Estimation procedure has resulted in the distinct mappings $M_1, M_2, \ldots, M_m$ with frequency proportions $f_1, f_2, \ldots, f_m$, respectively. Let $Q_{j,t}^{M_l(i_t)}$ be the index of product j corresponding to cluster $M_l(i_t)$. Note that the functions $a_{j,t}$ and $b_{j,t}$ are defined (and updated) separately for each cluster. Set product j's index as $Q_{j,t} = \sum_{l=1}^m f_l Q_{j,t}^{M_l(i_t)}$. Offer an assortment S_t that contains C products with the highest $Q_{j,t}$ indices.[5]

Remark 1. An alternative adaptation of Thompson Sampling would first randomly draw a mapping from $M_1, M_2, \ldots, M_m$ according to the frequency proportions $f_1, f_2, \ldots, f_m$, and then draw from the corresponding posterior Beta distributions. We favor our proposed approach because different mappings may only differ in terms of the composition of a few clusters. By averaging over different mappings, we take into account the similarities across different mappings. Our approach performs well, as evidenced in the case study.

5 Case Study

In this section, we discuss the results of several numerical experiments conducted on a dataset from a large Chilean retailer. We first provide a brief overview of the dataset in Section 5.1. We then discuss implementation details in Section 5.2. In Section 5.3, we compare the performance of the dynamic clustering policy to those of the data-intensive and linear-utility policies. We then study

[5] We can similarly adapt any index-based bandit policy, e.g., UCB1 of Auer et al. (2002). We obtained similar numerical results for UCB1 and therefore report only those based on Thompson Sampling in Section 5.3.

the impact of a finer representation of customer attributes (leading to a larger number of profiles) on the performance of policies in Section 5.4. We finally provide a more detailed comparison between the dynamic clustering policy and the linear-utility model in Section 5.5. The case study demonstrates the practical value of the dynamic clustering policy in a realistic setting. We find that the dynamic clustering policy outperforms the data-intensive and linear-utility policies as it benefits from pooling information and learning about customer preferences relatively faster. The case study also demonstrates the efficiency and scalability of the dynamic clustering policy in terms of computation time.

5.1 Dataset

The dataset that we use for the case study is from a chain of department stores owned by a Chilean multinational company headquartered in Santiago, Chile. The company sells clothing, footware, furniture, housewares, and beauty products, both through its network of department stores and through its online channel. In 2014, the company reported US$ 4.4 billion in gross profit. The dataset was collected as part of a larger field study by the retailer. In this study, the assortments offered to customers were chosen randomly without testing any assortment personalization strategy. The dataset consists of 94,622 customer-tied click records for a set of 19 products in the footware category (see Figure 2 below).[6] The dataset used in this study was collected through an experiment in which the 19 products were randomly assigned to 8 different assortments of 4 products each (i.e., $N = 19$ and $C = 4$). The experiment was conducted during a 32-day period through the retailer's online channel. Each arriving customer was shown one of these 8 assortments, chosen at random. Figure 2 illustrates an example of an assortment shown to customers. If the customer clicked on one of the products, that click was recorded in the dataset. Otherwise, a no-click was recorded. Therefore, each customer visit resulted in at most one click record. The dataset recorded the assortment history as well as the purchase/no-purchase decision (i.e., click/no-click decision in the context of the experiment) of each customer. The company uses information about the customers' location in Chile (according to a partition of the country into 7 different regions, determined by the retailer's marketing department: "Far North", "North," "Center," "South," "Far South," "Santiago West," and "Santiago East"), age group (the retailer uses three age groups, namely, $[0, 29]$, $[30, 39]$, and $[40, 99]$), and gender. This leads to a total of 42 unique vector of customer attributes. (The dataset actually contains more granular information on the age of customers, but we mostly use the age groups as determined by the Chilean retailer.)

5.2 Implementation Details

Estimation of Underlying Demand Model. We begin by using the dataset to estimate the parameters of the model. First, we estimate the distribution of customer arrivals (i.e., p_i) based on the number of transactions associated with each customer profile. Next, we estimate the underlying

[6] Each product is actually a "banner" that directs the customer to a page containing footware of that particular style/manufacturer – e.g., the second banner in Figure 2 leads to a page containing shoes with a 1970's style.

Figure 2: Example of an assortment shown on the retailer's website.

demand model from data. Because the set of available products belongs to the same category (shoes in this study), we use the MNL demand model as it accounts for product substitution. We also report on experiments based on the independent demand to explore the robustness of the results with respect to the underlying demand model.

For the MNL demand model, we estimate the exponentiated mean utility of each product for each customer profile separately. That is, we estimate 19 parameters for each of the 42 customer profiles using the transaction data only from that customer profile in the dataset. We do not assume any particular relation between the mean utilities and customer attributes when estimating the underlying demand model from data. Formally, for a profile i and product j, we estimate the exponentiated mean utility ν_j^i by

$$\hat{\nu}_j^i := \frac{\sum_u Z^i_{j,u}\, \mathbf{1}\{j \in S_u, i_u = i\}}{\sum_u Z^i_{0,u}\, \mathbf{1}\{j \in S_u, i_u = i\}}, \quad j \in \mathcal{N},\ i \in \mathcal{I}.$$

Note that, as in (6), the parameter estimation is conducted at the product level by exploiting the IIA property of the MNL model. For the case of independent demand, we estimate the purchase probability of each product for each profile as the sample mean of the number of purchases from data.

Data-Intensive Policy. We compare the performance of the dynamic clustering policy to that of the *data-intensive* policy, in which assortment decisions are made by treating each customer profile independently (as if each customer profile had a different distribution of preferences for products – even if this is not the case). Thus, under the data-intensive policy, the retailer assumes a deterministic mapping of customer profiles to clusters where each profile is mapped to a distinct cluster, i.e., $M(i) = i$ for $i \in I$. The data-intensive policy emphasizes the accuracy of preference estimation. That is, under this policy the retailer eventually learns the customers' preferences accurately, but at the expense of requiring a considerable amount of transaction data on each customer profile. Therefore, the data-intensive policy is prone to suffer from a slow learning speed.[7] We find that the dynamic clustering policy outperforms the data-intensive policy as a result of a

[7] We measure the speed of learning by evaluating the root mean squared error (between the actual and estimated parameters) over time.

faster learning speed achieved by pooling information across customers with similar preferences.

Linear-Utility Policy. We also compare the performance of the dynamic clustering policy to that of a policy that assumes a linear structure on the underlying demand model in terms of the dependence of utilities on customer attributes. We describe the MNL model in terms of the specific dataset available from the Chilean retailer. In particular, for this model, $x = (x_M, x_F, x_{A_1}, x_{A_2}, x_{A_3}, x_{L_1}, x_{L_2}, \ldots, x_{L_7})$ denotes the vector of attributes of a customer, where each variable is binary – x_M and x_F identify the gender of the customer (male and female, respectively), $x_{A_i},\ i \in \{1,2,3\}$ identifies the age group, and $x_{L_j},\ j \in \{1,\ldots,7\}$ identifies the location of the customer. The mean utility μ_j^x of a product j for a customer with profile x is assumed to take the form

$$\mu_j^x := \beta_j^\top x = \beta_j^0 + \beta_j^M x_M + \beta_j^{A_2} x_{A_2} + \beta_j^{A_3} x_{A_3} + \beta_j^{L_2} x_{L_2} + \beta_j^{L_3} x_{L_3} + \ldots + \beta_j^{L_7} x_{L_7}, \tag{7}$$

where β_j denotes the vector of coefficients, and β_j^0 captures the nominal utility of product j together with the effect of attributes "female", first age group (i.e., $[0,29]$), and Location 1. The vector of coefficient β_j is unknown to the retailer and must be estimated from the customers' transaction data. Similar to the data-intensive policy, the linear-utility policy treats each customer profile independently for optimization (i.e., following the same bandit algorithm). However, because this policy assumes that the underlying mean utilities of products are linear functions of customer attributes, the estimation of the β_j's is based on maximum likelihood estimation (MLE). Because different profiles might share some attributes, the MLE leverages information from similar profiles to estimate the preference parameters.

MCMC and Operating Machine. In order to estimate the mapping and customers' preferences under the dynamic clustering policy, we use $\eta = 300$ and a burn-in period of $\eta_b = 100$ iterations in the MCMC sampling scheme, after which every η_d =10th MCMC draw is used to estimate the mapping of profiles to clusters (this alleviates the auto-correlation between the MCMC draws). We also set the precision parameter of the Dirichlet Process to $\alpha = 1$. To expedite the computation time of the dynamic clustering policy, we set $\Phi = \{100, 200, 300, \ldots\}$; that is, we update the mapping of profiles to clusters every 100 customer arrivals. In between these periods, we use the prevailing mapping to update the preference parameters. (For consistency, we also update the estimates of the attribute-specific parameters in the linear-utility policy every 100 customer arrivals and use the prevailing estimates in between these periods.) All experiments were run on a machine with an Intel(R) i7-6700 3.40GHz CPU and 16GB of memory. In what follows we discuss the results of numerical experiments.

5.3 Performance Comparison

In this section, we compare the performance (in terms of regret) of the dynamic clustering policy to that of the data-intensive and linear-utility policies using the dataset from the Chilean retailer.

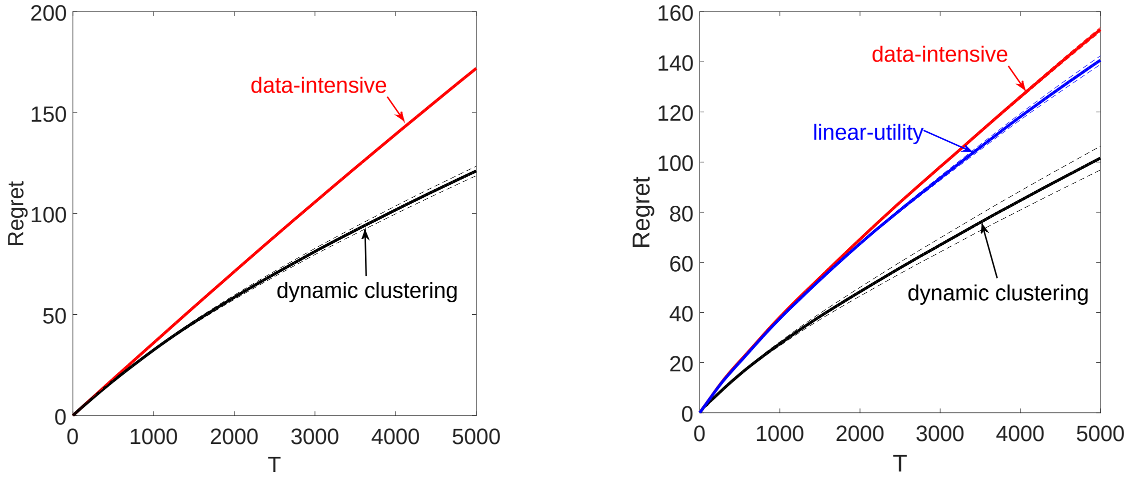

Figure 3: Average performance in a market with $I = 42$ profiles and independent demand (left), and MNL demand (right).

We consider the cases of both MNL and independent demands as the underlying demand models. We also compare the speed of learning between the dynamic clustering and data-intensive policies.

The underlying demand model and distribution of customer arrivals are estimated from the dataset. We run the experiments in markets with $T = 5000$ customers. There are nineteen products ($N = 19$) and the display constraint is of size four ($C = 4$). Moreover, there are $I = 42$ distinct customer profiles. We set prices $r_j = 1$ for all $j \in \mathcal{N}$. The reported performances are averaged over 100 replications and the dashed lines around a regret function represent the 95% confidence interval.

Figure 3 illustrates the average performance for the case of independent demand (left panel) and MNL demand (right panel). The dynamic clustering policy significantly outperforms the data-intensive and linear-utility policies as a result of pooling information. At the same time, the linear-utility policy outperforms the data-intensive policy as the latter suffers from a relatively slower learning speed.

Figure 4 (left panel) illustrates the evolution of the root mean squared error (RMSE) of estimated MNL parameters (i.e., exponentiated mean utilities) for the dynamic clustering and data-intensive policies in the case of the MNL demand model. This error is averaged over all products and profiles. As noted from the graph, the RMSE associated with the dynamic clustering policy decreases significantly faster than that of the data-intensive policy, implying a faster learning speed. The right panel of Figure 4 shows the evolution of the average number of clusters that emerge from the dynamic clustering policy over time (i.e., with the arrival of new customers). As can be noted from the graph, early on in the selling season, when only a limited number of transactions have been observed, the average number of clusters is small. That is, the policy pools transaction information across a large number of customer profiles. As more transaction data is collected, the dynamic clustering policy refines the composition of customer segments (clusters) and better personalizes the assortment offering using a larger number of clusters.

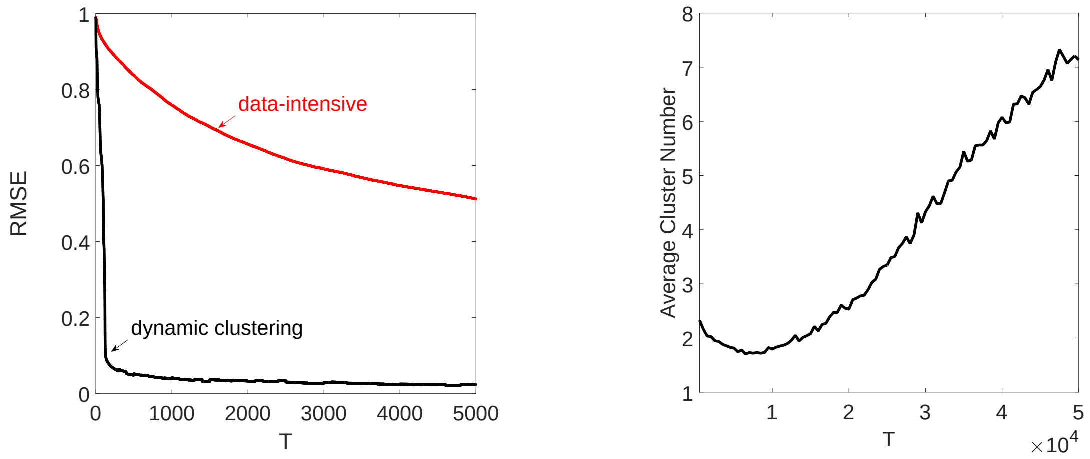

Figure 4: Speed of learning for different policies (left) and the evolution of the average number of clusters under the dynamic clustering policy (right) for MNL demand and $I = 42$.

The dynamic clustering policy outperforms the data-intensive and linear-utility policies in terms of regret (i.e., revenue collection) in the case study based on the Chilean retailer. We find that the dynamic clustering policy results (on average) in more than 37.7% and 27.3% additional transactions compared to the data-intensive and linear-utility policies, respectively. Moreover, the dynamic clustering policy results in more than 65% additional transactions compared to a randomized assortment policy (which was used by the retailer while collecting the data). Furthermore, the dynamic clustering policy has a significantly faster learning speed compared to the data-intensive policy. The proposed policy pools information across most profiles early on in the selling season, but personalizes the assortment offerings as more transaction data becomes available.

5.4 Customer Attributes

In this section, we study the impact of a finer definition of customer attributes on policy performance. Moreover, we compare the computation times of different policies and illustrate the efficiency and scalability of the dynamic clustering policy. In addition, we discuss an approach to further expedite the computation time of the dynamic clustering policy. We finally discuss how one can incorporate management knowledge about customer similarity into the dynamic clustering policy.

As discussed before, the Chilean retailer uses 42 different customer profiles, based on their understanding of the Chilean retail market. For example, the attribute corresponding to the customer's location is based on the retailer's knowledge about the customer base in Chile (e.g., urban versus rural locations, etc.). The retailer also groups customers according to their ages into three age groups $[0, 29]$, $[30, 39]$, and $[40, 99]$, presumably based on an understanding of different purchasing patterns of customers in each of these three groups. In order to study the impact of a finer definition of customer attributes, we have extended the study beyond the 42 original customer profiles used by the retailer. The raw dataset available to us contains more granular information

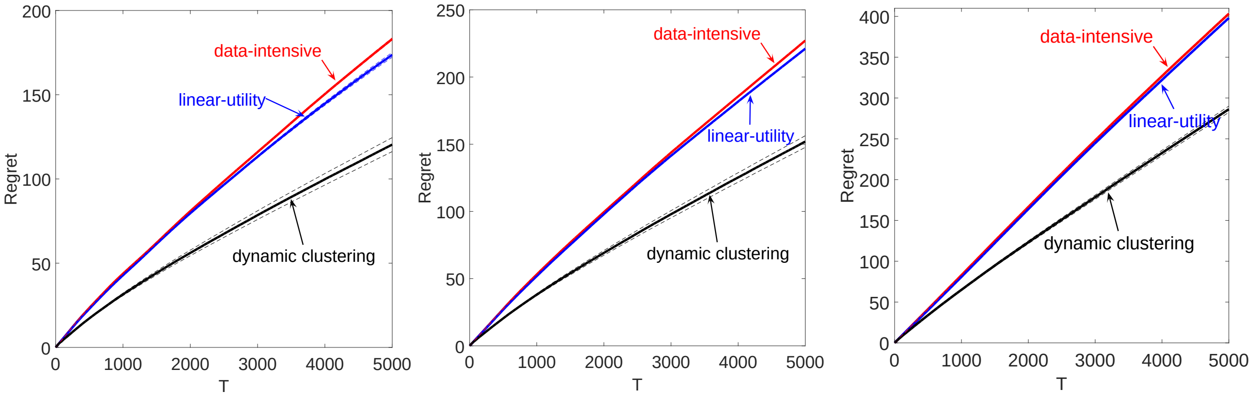

Figure 5: Average performance in a market with MNL demand and $I = 75$ (left), $I = 140$ (middle), and $I = 450$ (right).

about the customers' age (but not about the customers' location). As a result, we consider three additional studies that progressively refine the age attribute definition. These three studies consist of 75, 140, and 450 profiles, respectively. The first experiment with 75 distinct profiles uses age groups $[0, 30], [31, 40], [41, 50], \dots, [91, 99]$. The second one has 140 customer profiles using age groups $[0, 20], [21, 25], [26, 30], \dots, [96, 99]$. The third set of experiments has 450 customer profiles using exact ages $18, 19, 20, \dots, 99$. In each experiment, we only kept the profiles for which there was at least one no-purchase transaction in the dataset.

Figure 5 illustrates the average performance of the dynamic clustering, data-intensive, and linear-utility policies for the case of MNL demand. In all settings, the dynamic clustering policy outperforms the other policies in terms of regret (and thus revenue). That is, the better performance of the dynamic clustering policy relative to the other policies is robust with respect to the definition of customer attributes.

A finer set of customer attributes leads to an increased number of profiles, potentially slowing down the mapping estimation process and thus affecting computation times. Table 1 reports the computation time of all three policies for the cases of 42, 75, 140, and 450 customer profiles (all under the MNL demand model). The dynamic clustering policy requires the estimation of the mapping (by running the MCMC). In Table 1, we separate the running time of a customer arrival for which the mapping is updated (noted as MCMC) and the running time of those arrivals for which there is no mapping update (in which case the policy uses the prevailing mapping of profiles to clusters). Similarly, the linear-utility model requires the estimation of attribute-specific parameters through the maximum likelihood estimation (MLE). Table 1 reports the running time of a customer arrival for which the attribute-specific parameters are updated (noted as MLE) and the running time of those arrivals in which there is no parameter update (and for which the linear-utility policy operates under the prevailing estimates). The computation time of the MCMC and MLE increases with the granularity of customer attributes (and therefore the number of customer profiles).

The computation time of the dynamic clustering policy is reasonable and scales well with the number of profiles. All experiments were run on a personal computer – a retailer with more sophisticated computational resources would experience even faster results. Note that, as expected, the data-intensive policy has the fastest computation time (at the expense of a lower revenue performance), as this policy treats each profile independently. The dynamic clustering policy can also handle a large number of products without significantly affecting computation times. In the estimation stage, the number of products impacts the calculation of the likelihood functions and the updates of preference parameters, which are computed in closed-form as discussed in Section 4.3. For the assortment optimization stage, there are efficient algorithms for the MNL model (for example, the algorithm in Rusmevichientong et al. (2010) scales polynomially in the number of products). The optimization in the case of independent demand is trivial.

	Dynamic Clustering		Linear-Utility		Data-Intensive
	No mapping update	MCMC	No parameter update	MLE	Each customer arrival
$I = 42$	0.00031	1.0818	0.00017	2.7827	0.00015
$I = 75$	0.00039	1.7121	0.00018	3.9762	0.00015
$I = 140$	0.00054	2.9850	0.00018	5.9750	0.00016
$I = 450$	0.00117	8.1729	0.00018	9.1619	0.00017

Table 1: Average computation time (in seconds) of different policies for MNL demand and $T = 5000$.

We next discuss an approach to further improve the computation time of the dynamic clustering policy, without significantly impacting its performance. This approach involves reducing the number of customer profiles for clustering purposes. In particular, this version of the dynamic clustering policy considers customer profiles within each specific location in isolation and therefore the policy runs the MCMC for different locations in parallel. We refer to this version of the dynamic clustering policy as “Location-DC.” Table 2 below reports the computation times of this version of the dynamic clustering policy, together with that of the original policy (which we refer to as “Original-DC”), for the setting with $I = 450$ profiles and MNL demand. As can be noted from the table, the “Location-DC” version of the policy brings significant savings in terms of computation time. While this version is slightly outperformed by the original dynamic clustering policy in terms of regret, it still performs significantly better than the data-intensive and linear-utility policies.

	No mapping update	MCMC
Original-DC	0.00117	8.1729
Location-DC	0.00048	1.5575

Table 2: Average computation time (in seconds) of the original and a location-based version of the dynamic clustering policy for MNL demand with $I = 450$ and $T = 5000$.

We finally discuss how one can incorporate management knowledge about customer similarity into the dynamic clustering policy. Suppose that existing management insight indicates that “neighboring” profiles (e.g., two profiles with the same gender and location and very close in age) are likely to have “similar” preferences for products. The dynamic clustering algorithm can accommodate management knowledge about customers by restricting attention to mappings that only group "neighboring" profiles. (This can be implemented in the MCMC sampling procedure: while updating the cluster label c_i^* of a customer with profile given by a vector x^i, the candidate cluster label is drawn only from clusters that currently include profiles that are "similar" to x^i – the notion of similarity can be formally defined based on attributes.) An alternative approach consists of merging upfront profiles that are known to have similar preferences for products. For example, it may be that, based on an understanding of the customer base, management is confident that customers from two particular ZIP codes or locations have similar tastes for products. Unlike the previous approach, these customer profiles with similar preferences may not have similar attributes. One can use this additional information to speed up the learning process by grouping such profiles together before running the dynamic clustering algorithm.

In sum, we find that the better performance of the dynamic clustering policy relative to the other policies is robust with respect to the definition of customer attributes. At the same time, a finer definition of customer attributes (i.e., a larger number of customer profiles) increases the computation time. However, we show that the computation time of the dynamic clustering policy is still reasonable and scales well with the number of profiles. One can further expedite the computation time of the policy by reducing the number of customer profiles for clustering purposes.

5.5 Comparison to Linear-Utility Model

In this section, we compare the dynamic clustering policy to the linear-utility approach in more detail. We first discuss the advantages of each approach and then introduce a set of experiments to compare the performance of the two policies.

In addition to better performance of the dynamic clustering policy over the linear-utility model, the dynamic clustering policy has several other advantages. Retailers are generally interested in identifying customer segments (i.e., clusters of customers with similar preferences). These segments are part of the output of the dynamic clustering policy and can be interpreted based on customers' attributes – see Figure 6 below for a representative example based on the dataset from the Chilean retailer. Figure 7 also illustrates the optimal assortments for different clusters in an example in which the profiles only differ by their geographical location in Chile. Moreover, the mean utilities may not be linear in customer attributes. The dynamic clustering policy makes no assumption about the structure of the mean utilities with respect to customer attributes. In particular, the transaction dataset from the Chilean retailer exhibits a non-linear dependence between mean utilities and attributes. In such cases, the linearity assumption could result in inaccurate estimates which, in turn, could hurt the retailer's revenue. Also, the dynamic clustering policy takes a Bayesian semi-parametric approach and is designed to expedite the learning process, especially in the short-term when the amount of transaction data is limited. Overall, the dynamic clustering policy leads to 27.3% more transactions (on average) than the linear-utility policy in the experiments based on the dataset from the Chilean retailer. The linear-utility approach, however, uses maximum-likelihood estimation and thus is better suited for (offline) settings with large amounts of transaction data. As such, the linear-utility approach can identify whether an attribute is statistically significant (relevant) through the estimates of the attribute-specific parameters. One can make similar observations based on the output of the dynamic clustering policy. For example, as noted in Figure 6, customers in Santiago West tend to have different preferences for products than customers from Santiago East.

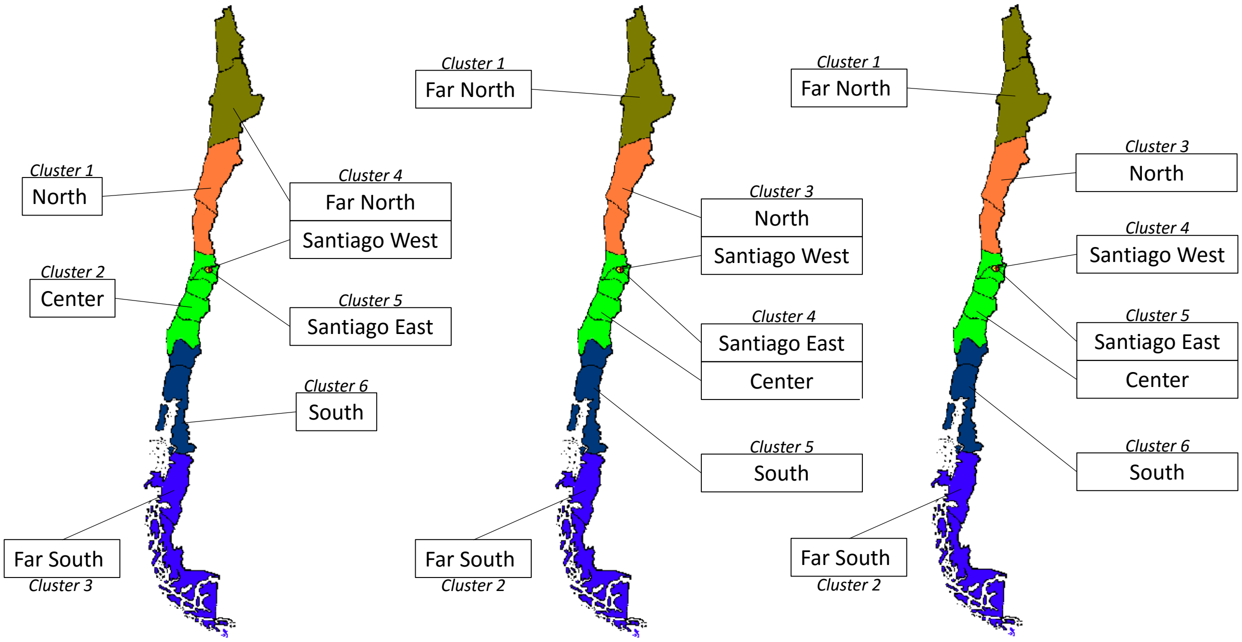

Figure 6: Illustration of clusters (under the most likely mapping) for women from age groups $[0, 29]$ (left), $[30, 39]$ (middle), and $[40, 99]$ (right) based on customers' location in Chile.

Because the linear-utility approach may be better suited for an offline setting, we designed an additional set of experiments which mimics an offline setting. More specifically, because the estimation approach (and underlying model assumption) is different under the dynamic clustering and linear-utility policies, we further compare the performance of these policies in a setting in which the performance is mainly affected by the quality of estimation which, in turn, impacts the assortment decisions made by different policies. To that end, we consider separation-based versions of these policies, which separate exploration from exploitation. In the separation-based experiments, we randomly generate a sample of transaction data with random assortment offerings and random customer arrivals based on the estimated MNL parameters and arrival distribution of profiles from the dataset. All policies use the same sample to estimate the MNL parameters (exploration stage). Each policy then finds the optimal assortment for each customer profile. We then generate another random sample of customer arrivals with the same size as that used in the exploration phase (according to the estimated arrival distribution) over which each policy offers its personalized optimal assortment to each arriving customer (exploitation stage). We consider the cumulative regret and revenue of all policies only for the exploitation phase as all policies use the same sample (and thus incur the same regret) in the exploration phase. As a result, any difference in performance is due to the quality of estimation. We consider two scenarios in terms of the

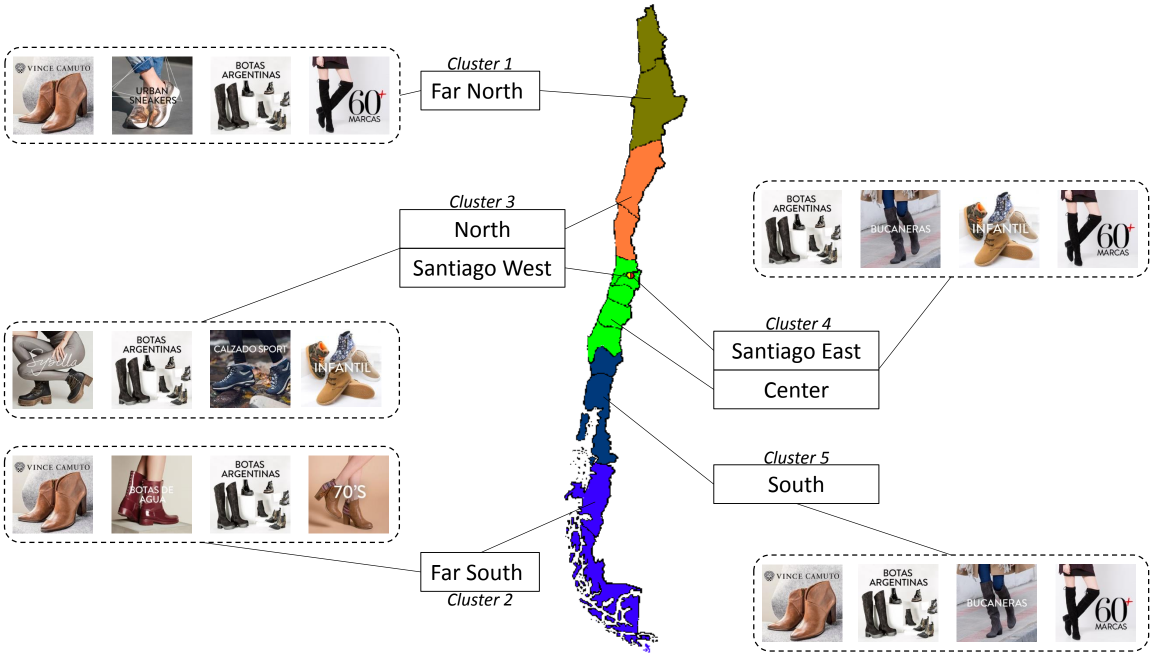

Figure 7: Illustration of optimal assortments for women from age group $[30, 39]$.

underlying demand model.

The first scenario is based on the original transaction dataset, which exhibits a non-linear dependence of mean utilities on customer attributes. We experiment with different sample sizes T. The dynamic clustering policy outperforms both the data-intensive and linear-utility policies in all instances. Table 3 below reports the percentage improvement of the dynamic clustering policy over the other policies, both in terms of regret and revenue (i.e., expected number of transactions). The revenue improvements can be as high as 50% and 25.4% compared to the data-intensive and linear-utility policies, respectively. Moreover, the improvements are generally higher for smaller samples sizes, as the dynamic clustering policy performs particularly well in settings with limited transaction data by pooling information. Table 4 reports the average estimation times (i.e., computation time of MCMC and MLE for the dynamic clustering and linear-utility policies, respectively). As expected, the data-intensive policy has the lowest computation time as the estimation can be done independently for each customer profile. The linear-utility model is faster than the dynamic clustering policy for smaller sample sizes. However, its computation time increases significantly for larger sample sizes, suggesting that the linear-utility policy may be better suited for an offline setting.

While the original dataset does not exhibit a linear dependence on customer attributes, we generate a second set of (synthetic) experiments in which the underlying demand model is linear as in (7). To that end, we estimate β_j in model (7) for each product j from the dataset and use such estimates to randomly generate synthetic (linear) transaction data for simulation. We find that the dynamic clustering policy outperforms the linear-utility approach in all instances except

	Regret		Revenue	
	Data-Intensive	Linear-Utility	Data-Intensive	Linear-Utility
$T = 500$	34.2%	24.6%	47.8%	25.4%
$T = 1000$	42.6%	28.4%	50.0%	21.6%
$T = 5000$	39.0%	26.6%	20.7%	10.7%
$T = 10000$	24.3%	12.9%	8.9%	3.9%
$T = 20000$	6.7%	3.6%	1.5%	0.8%

Table 3: Average percentage improvement of dynamic clustering policy over data-intensive and linear-utility policies in separation-based experiments using the original dataset.

	$T = 500$	$T = 1000$	$T = 5000$	$T = 10000$	$T = 20000$
Dynamic Clustering	1.214	1.301	2.307	3.861	7.010
Data-Intensive	0.003	0.006	0.025	0.048	0.096
Linear-Utility	0.508	0.922	5.872	13.877	31.168

Table 4: Average estimation time (in seconds) in separation-based experiments based on original dataset.

for that with the largest transaction sample (i.e., $T = 20000$). This suggests that the benefit of pooling information achieved by clustering can lead to better performance even when the underlying demand model is, in fact, linear. Such benefits are more pronounced in the short-term, i.e., for small and moderate sample sizes.

The dynamic clustering policy outperforms the linear-utility approach in the case study with the Chilean retailer's dataset. Each approach has its advantages and may be deemed more appropriate depending on the specific setting. For example, the Chilean retailer that provided the dataset is interested in identifying customer segments (i.e., clusters of customers with similar preferences). These segments are part of the output of the dynamic clustering policy and can be interpreted based on customers' attributes (as in the examples in Figures 6 and 7).

6 Value of Pooling Information

In this section, we provide analytical support for the insights derived in the case study in Section 5. More specifically, this section explores the impact of pooling information about customers' preferences on the retailer's revenue by considering a stylized version of the dynamic assortment personalization problem. To this end, we focus on three policies that differ in the extent by which they aggregate information across customers. The data-intensive policy, described in Section 5.2, treats customer profiles independently to estimate preferences and make assortment decisions. We further introduce a *semi-oracle* policy that knows upfront the underlying mapping of profiles to clusters but not the customer preferences for each cluster. The semi-oracle policy reflects the key element of the dynamic clustering policy – in that it pools transaction information across customers with similar preferences – but it bypasses the estimation of the mapping of profiles to clusters by assuming that it is known to the retailer. Working with the dynamic clustering policy analytically is not possible, as it requires a Bayesian update of the mapping that cannot be done in closed-form. In the other extreme, we consider a *pooling* policy that aggregates transaction data across *all* customer profiles (regardless of whether the customers have similar preferences or not). We show that the semi-oracle outperforms the data-intensive policy. Moreover, we analytically characterize settings in which the pooling policy outperforms the data-intensive policy.

All policies we consider in this section – the data-intensive policy ($\pi^{d\text{-}int}$), the semi-oracle policy ($\pi^{s\text{-}orc}$), and the pooling policy (π^{pool}) – follow the same bandit algorithm to determine what assortment to offer each arriving customer. The policies, however, differ in how they use the available information to estimate the customers' preferences and make assortment decisions. Thus, the results in this section provide insights about the benefit of pooling information by analytically exploring a simplified version of the problem. For ease of analysis, we focus on the *independent* demand model in this section.[8] We also assume, for tractability, that $C = 1$ and $r_j = 1$ for all products $j \in \mathcal{N}$.

Let $R^{\pi^{d\text{-}int}}$, $R^{\pi^{s\text{-}orc}}$, and $R^{\pi^{pool}}$ denote the regrets associated with the data-intensive, semi-oracle, and pooling policies, respectively. To simplify notation, we denote them as $R_{d\text{-}int}$, $R_{s\text{-}orc}$, and R_{pool} hereafter. We further define the gap functions

$$G_1 := R_{d\text{-}int} - R_{s\text{-}orc} \quad \text{and} \quad G_2 := R_{d\text{-}int} - R_{pool}.$$

Our goal is to determine conditions under which these gaps are non-negative. Because characterizing the regret functions in closed-form is not possible, we use upper bounds on the regret for the semi-oracle and pooling polices, denoted by $U_{s\text{-}orc}$ and U_{pool}, respectively, and a lower bound on the regret for the data-intensive policy, denoted by $L_{d\text{-}int}$. Therefore, $L_{d\text{-}int} - U_{s\text{-}orc}$ provides a lower bound for the gap function G_1 and $L_{d\text{-}int} - U_{pool}$ provides a lower bound for the gap function G_2. Hence, we focus on characterizing settings in which these lower bounds are non-negative, which in turn implies that $G_1, G_2 \geq 0$.[9] Lai and Robbins (1985) prove an asymptotic lower bound (for large T) on the achievable performance of any *consistent* policy in the classic bandit setting (i.e., with a homogeneous population of customers).[10] Roughly speaking, the long-run number of mistakes (associated with pulling suboptimal arms) under any consistent policy is smaller than T^a for large T and every $a > 0$. In particular, it is smaller than a linear function of T, which corresponds to making mistakes for every customer. Let $\mathcal{P}' \subseteq \mathcal{P}$ denote the set of consistent admissible policies. We restrict attention to consistent policies $\pi \in \mathcal{P}'$ and use Lai and Robbins' lower bound to derive $L_{d\text{-}int}$. More specifically, we derive a lower bound on the regret associated with each profile and define $L_{d\text{-}int}$ as the sum of these lower bounds.

The upper bound on the regret for the semi-oracle and pooling policies depend on the specific bandit algorithm used for selecting the product to offer each arriving customer. We focus here on

[8] One can obtain similar results for the MNL demand model as well.

[9] While the lower bounds are not always tight, the goal is to show the non-negativity of the gap functions. As a result, working with the bounds enables the analysis and leads to the desired results.

[10] An admissible policy π is consistent if, for any distribution of preferences F (that satisfies certain regularity conditions), $\frac{R^{\pi}(T)}{T^a} \to 0$, as $T \to \infty$, for every $a > 0$. That is, if $R^{\pi}(T) = o(T^a)$. See Lai and Robbins (1985).

the celebrated upper confidence bound (UCB1) policy of Auer et al. (2002). After an initialization phase during which each product is offered once, UCB1 offers customer t a product j with the highest index $\bar{\mu}_j + \sqrt{2\ln(t-1)/k_j(t-1)}$, where $\bar{\mu}_j$ is the sample mean of the number of purchases for product j, and $k_j(t-1)$ is the number of times that product j has been offered up to (and including) time $t-1$. The UCB1 policy is easy to implement and its regret admits a finite-time upper bound which is simple to use. We extend the results of Sections 6.1 and 6.2 for Thompson Sampling in Appendix B.

6.1 Semi-Oracle

In this section, we compare the performance of the data-intensive policy to that of the semi-oracle policy. As expected, the semi-oracle outperforms the data-intensive policy in terms of regret (i.e., revenue). This result emphasizes the benefit of estimating the mapping of profiles to clusters as it helps expedite the learning process by pooling transaction information across customer profiles within a cluster. (We provide a formal statement and proof of the result in Theorem A.1 in Appendix A.)

We next show that there are diminishing marginal returns to pooling information from an increasing number of customer profiles. To that end, consider a general market with K clusters where $1 \leq K < I$. We assume, without loss of generality, that $K < I$, since if $K = I$, then both the semi-oracle and data-intensive policies incur the same regret and therefore $G_1 = 0$. Let $\mathcal{I}_k$ denote the set of profiles belonging to cluster k and $I_k := |\mathcal{I}_k|$. Also, let $\mathcal{I}' := (\mathcal{I}_1, \ldots, \mathcal{I}_K)$. This vector summarizes the mapping of profiles to clusters. We assume, without loss of generality, that product 1 has the highest purchase probability for each profile, i.e., $\mu_j^i < \mu_1^i$ for $j = 2, \ldots, N$ and all $i \in \mathcal{I}$. Computing the lower bound $L_{\textit{d-int}} - U_{\textit{s-orc}}$ for the gap function G_1 requires an additional approximation as the setting studied in Lai and Robbins (1985) considers a homogeneous population of customers. This additional approximation involves a first-order Taylor expansion and, as such, the resulting approximate lower bound is very close to $L_{\textit{d-int}} - U_{\textit{s-orc}}$. We denote by $G_{1l}(T, \mathcal{I}')$ the approximation to the lower bound for the gap function G_1. This approximate lower bound depends on the total number of customer arrivals T and on the vector $\mathcal{I}'$ which encodes the mapping of profiles to clusters. We provide a detailed derivation of $G_{1l}(T, \mathcal{I}')$ in Appendix A.

Theorem 1. *Consider the case of uniform arrivals within each cluster, i.e., $p_i = P_k/I_k$ for all $i \in \mathcal{I}_k$ where $P_k := \sum_{i \in \mathcal{I}_k} p_i$ is constant. We then have that $G_{1l}(T, \mathcal{I}')$ is increasing in I_k for $T > eI_k/P_k$ (e is the Euler's number) and concave in I_k for $T \geq 1$.*

Theorem 1 shows the first- and second-order effects of the number of customer profiles I_k on the approximate lower bound. We find that, for sufficiently large T, $G_{1l}(T, \mathcal{I}')$ is increasing in the number of profiles I_k. That is, the benefit of pooling information increases with the number of profiles within any cluster k, as it becomes increasingly more time-consuming for the data-intensive policy to learn the preferences of each customer profile when I_k increases. In addition, the result shows that $G_{1l}(T, \mathcal{I}')$ is concave in I_k. That is, there are diminishing marginal returns to pooling information from an increasing number of customer profiles within any cluster.

We also explore whether the result in Theorem 1 applies to the dynamic clustering policy and MNL demand model by considering an example based on the dataset from the Chilean retailer. Specifically, we estimate the MNL parameters for the customer profile (female, $[40, 99]$, Center) from the dataset and assume that a cluster's demand follows such MNL model. We then increase the number of profiles in that cluster and evaluate the gap between the regrets of the data-intensive and dynamic clustering policies in a market with $T = 5000$ customers. Figure 8 shows the result of this experiment, where $R_{d\text{-}int}$ and R_{dc} denote the regrets of data-intensive and dynamic clustering policies, respectively. As noted in the graph, and consistent with Theorem 1, the (actual) gap between the regrets of the two policies is increasing and concave in the number of customer profiles within the cluster.

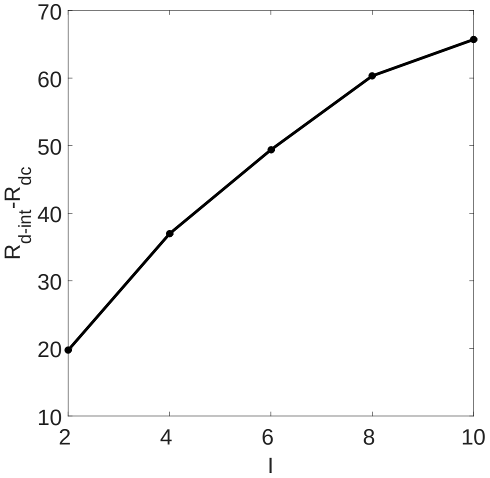

Figure 8: Gap between the regrets of data-intensive and dynamic clustering policies as a function of number of profiles.

6.2 Pooling in the Short-Term

In this section, we consider a setting with heterogeneous customers and a *pooling* policy that aggregates information across *all* customer profiles. As one would expect, pooling information across all customer profiles is not necessarily beneficial for the retailer in a heterogeneous market as it could lead to erroneous estimates. However, we show that, under some conditions, the pooling policy tends to outperform the data-intensive policy in the short-term even if customer preferences are heterogeneous. This, in turn, allows us to examine the key drivers of efficiency gains derived by pooling information.

Consider a market with $K \geq 2$ clusters. Without loss of generality, we assume that $K = N$, where N is the number of products. We also assume that cluster k's customers have the highest purchase probability for product k, for $k = 1, \dots, K$. Furthermore, we assume that $\mu_k^k - \mu_j^k = \Delta$ for some $\Delta > 0$ and for all $k = 1, \dots, K$ and $j \neq k$, where, to simplify notation, μ_j^k denotes the purchase probability of product j for all profiles in cluster k. Let $P := (P_1, P_2, \dots, P_K)$ where $P_k = \sum_{i \in \mathcal{I}_k} p_i$

is the proportion of profiles belonging to cluster k. We also define $P' := (P_2, \ldots, P_{K-1})$. We assume, without loss of generality, that $P_k < P_1 \leq 1$ for all $k = 2, \ldots, K$. Note that P_1 is a measure of heterogeneity in this setting and a smaller P_1 leads to a more heterogeneous customer population. In the extreme, $P_1 = 1$ reduces this setting to one with a homogeneous market.

As in Section 6.1, we consider an approximate lower bound on the regret of the data-intensive policy, based on Lai and Robbins (1985). Existing upper bounds in the bandit literature (including that of the UCB1 policy) assume a homogeneous market. This introduces additional complexity in the computation of the upper bound on the regret of the pooling policy. This upper bound is derived in Appendix A. Let $G_{2l}(t, I, P)$ be the approximation of the lower bound $L_{d\text{-}int} - U_{pool}$ to the gap function G_2 at any time period t. We provide a detailed derivation of $G_{2l}(t, I, P)$ in Appendix A. The next result provides conditions under which the pooling policy outperforms the data-intensive policy (subject to the approximations), that is, $G_{2l}(t, I, P) \geq 0$.

Theorem 2. *Consider the case of uniform arrivals, i.e., $p_i = 1/I$ for all $i \in \mathcal{I}$. There exist thresholds $\tilde{I}_l(P)$ and $\tilde{P}_1(I, P')$ such that if $I \geq \tilde{I}_l(P)$ and $\tilde{P}_1(I, P') < P_1 \leq 1$, then*

$$G_{2l}(t, I, P) \geq 0 \quad \textit{for} \quad \tilde{t}_l(I, P) \leq t \leq \tilde{t}_u(I, P),$$

with $1 < \tilde{t}_l(I, P) \leq \tilde{t}_u(I, P) \leq \infty$. Moreover, $\tilde{I}_l(P)$ and $\tilde{P}_1(I, P')$ are non-increasing in P_1 and I, respectively.

The result in Theorem 2 shows that, under some conditions, the pooling policy outperforms the data-intensive policy (subject to the approximations) for a range of customer arrivals, even if the retailer learns the customers' preferences inaccurately in a heterogeneous market under the pooling policy. This is the result of faster learning under the pooling policy achieved by aggregating information across all customer profiles. In particular, Theorem 2 illustrates the benefit of pooling information in the short-term, when transaction data is limited. Moreover, Theorem 2 implies that three key factors favor the performance of the pooling policy over the data-intensive policy:

- *Heterogeneity* (P_1): If the population is not too heterogeneous (i.e., if $P_1 > \tilde{P}_1(I, P')$), then the pooling policy tends to outperform the data-intensive policy for a range of customer arrivals. This is because, under such condition, the benefit associated with faster learning by aggregating information outweighs the cost associated with the errors the pooling policy makes by not differentiating between clusters (and therefore offering suboptimal assortments). Moreover, the threshold $\tilde{P}_1(I, P')$ decreases as the number of profiles increases.

- *Number of profiles* (I): An increase in the number of profiles impacts negatively on the performance of the data-intensive policy. As I increases, the average number of customer arrivals per profile decreases and thus it takes longer for the data-intensive policy to learn the preferences for each profile. On the other hand, the pooling policy aggregates information across all customers and thus its performance does not degrade as long as $P_1 > \tilde{P}_1(I, P')$. As the population becomes more homogeneous in terms of preferences (i.e., as P_1 increases), the pooling policy tends to outperform the data-intensive policy for an even smaller number of customer profiles.

- *Number of Customers* (t): Although a relatively more homogeneous market and a large number of profiles can favor the performance of the pooling policy, the *key* factor is the amount of transaction information available to the retailer. When the number of customer arrivals is still relatively small, the data-intensive policy does not have enough sample points to accurately learn the preference of each customer profile. On the other hand, the pooling policy aggregates information and therefore tends to outperform the data-intensive policy as long as the population is not too heterogeneous with respect to their product preferences. As more customers arrive, the performance of the data-intensive policy prevails. In particular, $G_{2l}(t, I, P) \to -\infty$ (and $G_2(t, I, P) \to -\infty$) as $t \to \infty$ if $P_1 < 1$.

The result in Theorem 2 is consistent with the observations about the dynamic clustering policy illustrated in the right panel of Figure 4. Early on in the selling season, when only a limited number of transactions have been observed, the average number of clusters that emerge from the dynamic clustering policy is small. This echoes the preceding discussion – when limited information is available, the retailer might be better off pooling all available data (even if they correspond to profiles with different preferences) to speed up the learning process. The number of clusters then increases as more data becomes available (as can be noted in the right panel of Figure 4) and the retailer is able to personalize the assortment offerings by better matching customer preferences.

7 Conclusion

This paper considers a retailer endowed with multiple products that dynamically personalizes the assortment offerings over a finite selling season. Customers are assigned to different profiles based on their observable personal attributes. Their preferences are unknown to the retailer and must be learned over time. The primary goal of the paper is to explore the efficient use of data in retail operations and its benefits (in terms of revenue) for assortment personalization. To that end, we propose the *dynamic clustering* policy as a prescriptive approach for assortment personalization in an online setting. We take advantage of existing tools from the literature and introduce a policy that adaptively combines estimation (by estimating customer preferences through dynamic clustering) and optimization (by making dynamic personalized assortment decisions using a bandit policy) in an online setting. The dynamic clustering policy adaptively adjusts the composition of customer segments (i.e., mapping of profiles to clusters) based on the observed customers' purchasing decisions. The policy exploits the similarity in preferences of customers in the same cluster by aggregating their transaction information and expediting the learning process. Using the estimated mapping and preferences, the policy uses existing bandit algorithms to make assortment decisions.

To illustrate the practical value of the dynamic clustering policy in a realistic setting, we apply the policy to a dataset from a large Chilean retailer. We compare the performance of the dynamic clustering policy with two alternatives: a data-intensive policy that treats each customer profile independently and a linear-utility policy that estimates product mean utilities as linear functions of customer attributes. The case study suggests that the dynamic clustering policy can significantly increase the average number of transactions relative to the other policies. We also demonstrate the scalability and efficiency of the dynamic clustering policy in terms of computation time.

We then study a simplified version of the problem in which the retailer offers a single product to each arriving customer. We show that a semi-oracle policy that knows upfront the mapping of profiles to clusters (but not the customers' preferences) outperforms the data-intensive policy, indicating that pooling information is beneficial for the retailer. We also demonstrate that there are decreasing marginal returns to pooling information as the number of customer profiles increases. Finally, we characterize conditions under which a policy that pools information across all customer profiles outperforms the data-intensive policy even when customer preferences are different. This result emphasizes the benefit of pooling information in the short-term, when there is insufficient data to accurately estimate preferences for each customer profile.

In this work, we have made some simplifying assumptions for tractability. Future work can take into consideration the presence of inventory constraints in the model. Moreover, we have assumed that prices are constant throughout the selling season. Incorporating pricing decisions is another direction for future research.

References

Agrawal, S., Avadhanula, V., Goyal, V. and Zeevi, A. (2016). MNL-bandit: A dynamic learning approach to assortment selection. Working paper, Columbia University.

Agrawal, S. and Goyal, N. (2011). Analysis of thompson sampling for the multi-armed bandit problem. *arXiv preprint arXiv:1111.1797* .

Anantharam, V., Varaiya, P. and Walrand, J. (1987). Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays-part i: IID rewards. *Automatic Control, IEEE Transactions on* **32**(11), 968–976.

Ansari, A. and Mela, C. F. (2003). E-customization. *J. Marketing Res.* **40**(2), 131–145.

Antoniak, C. E. (1974). Mixtures of dirichlet processes with applications to bayesian nonparametric problems. *The annals of statistics* pp. 1152–1174.

Arora, N., Dreze, X., Ghose, A., Hess, J. D., Iyengar, R., Jing, B., Joshi, Y., Kumar, V., Lurie, N., Neslin, S. et al. (2008). Putting one-to-one marketing to work: Personalization, customization, and choice. *Marketing Lett.* **19**(3-4), 305–321.

Auer, P., Cesa-Bianchi, N. and Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine Learn.* **47**(2-3), 235–256.

Bernstein, F., Kök, A. G. and Xie, L. (2015). Dynamic assortment customization with limited inventories. *Manufacturing Service Oper. Management* **17**(4), 538–553.

Besbes, O. and Sauré, D. (2016). Product assortment and price competition under multinomial logit demand. *Production Oper. Management* **25**(1), 114–127.

Besbes, O. and Zeevi, A. (2009). Dynamic pricing without knowing the demand function: Risk bounds and near-optimal algorithms. *Oper. Res.* **57**(6), 1407–1420.

Burda, M., Harding, M. and Hausman, J. (2008). A bayesian mixed logit–probit model for multinomial choice. *Journal of Econometrics* **147**(2), 232–246.

Caro, F. and Gallien, J. (2007). Dynamic assortment with demand learning for seasonal consumer goods. *Management Sci.* **53**(2), 276–292.

Chen, X., Owen, Z., Pixton, C. and Simchi-Levi, D. (2015). A statistical learning approach to personalization in revenue management. *Available at SSRN 2579462* .

Cheung, W. C., Simchi-Levi, D. and Wang, H. (2016). Dynamic pricing and demand learning with limited price experimentation. Working paper, MIT.

Ciocan, D. F. and Farias, V. F. (2014). Fast demand learning for display advertising revenue management. Working paper, MIT.

eMarketer (2014). Global b2c ecommerce sales to hit $1.5 trillion this year driven by growth in emerging markets. http://www.emarketer.com/Article/Global-B2C-Ecommerce-Sales-Hit-15-Trillion-This-Year-Driven-by-Growth-Emerging-Markets/1010575.

Ferguson, T. (1973). A bayesian analysis of some nonparametric problems. *The annals of statistics* pp. 209–230.

Ferguson, T. S. (1983). Bayesian density estimation by mixtures of normal distributions. *Recent advances in statistics* **24**, 287–302.

Ferreira, K. J., Simchi-Levi, D. and Wang, H. (2016). Online network revenue management using thompson sampling. Working paper, Harvard Business School.

Fisher, M. and Vaidyanathan, R. (2014). A demand estimation procedure for retail assortment optimization with results from implementations. *Management Sci.* **60**(10), 2401–2415.

Gallego, G., Li, A., Truong, V.-A. and Wang, X. (2015). Online resource allocation with customer choice. *arXiv preprint arXiv:1511.01837* .

Gelman, A., Carlin, J. B., Stern, H. S. and Rubin, D. B. (2014). *Bayesian data analysis*. Vol. 2. Taylor & Francis.

Golrezaei, N., Nazerzadeh, H. and Rusmevichientong, P. (2014). Real-time optimization of personalized assortments. *Management Sci.* **60**(6), 1532–1551.

Harrison, J. M., Keskin, N. B. and Zeevi, A. (2012). Bayesian dynamic pricing policies: Learning and earning under a binary prior distribution. *Management Sci.* **58**(3), 570–586.

Heller, K. A. and Ghahramani, Z. (2005). Bayesian hierarchical clustering. 'Proceedings of the 22nd international conference on Machine learning'. ACM. pp. 297–304.

Jasin, S. and Kumar, S. (2012). A re-solving heuristic with bounded revenue loss for network revenue management with customer choice. *Mathematics of Operations Research* **37**(2), 313–345.

Kallus, N. and Udell, M. (2016). Dynamic assortment personalization in high dimensions. *arXiv preprint arXiv:1610.05604* . Working paper, Cornell University.

Kök, A. G., Fisher, M. L. and Vaidyanathan, R. (2015). Assortment planning: Review of literature and industry practice. *Retail Supply Chain Management* pp. 175–236.

Lai, T. L. and Robbins, H. (1985). Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics* **6**(1), 4–22.

Linden, G., Smith, B. and York, J. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. *Internet Computing, IEEE* **7**(1), 76–80.

Modaresi, S., Sauré, D. and Vielma, J. P. (2014). Learning in combinatorial optimization: What and how to explore. Working paper, Duke University.

Montgomery, A. L. and Smith, M. D. (2009). Prospects for personalization on the internet. *Journal of Interactive Marketing* **23**(2), 130–137.

Murthi, B. and Sarkar, S. (2003). The role of the management sciences in research on personalization. *Management Sci.* **49**(10), 1344–1362.

Neal, R. (2000). Markov chain sampling methods for dirichlet process mixture models. *Journal of Computational and Graphical Statistics* **9**(2), 249–265.

Robbins, H. (1985). Some aspects of the sequential design of experiments. *Herbert Robbins Selected Papers* pp. 169–177.

Rusmevichientong, P., Shen, Z.-J. M. and Shmoys, D. B. (2010). Dynamic assortment optimization with a multinomial logit choice model and capacity constraint. *Oper. Res.* **58**(6), 1666–1680.

Sauré, D. and Zeevi, A. (2013). Optimal dynamic assortment planning with demand learning. *Manufacturing Service Oper. Management* **15**(3), 387–404.

Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika* pp. 285–294.

Ulu, C., Honhon, D. and Alptekinoğlu, A. (2012). Learning consumer tastes through dynamic assortments. *Oper. Res.* **60**(4), 833–849.

Wedel, M. and Kamakura, W. A. (2012). *Market segmentation: Conceptual and methodological foundations.* Vol. 8. Springer Science & Business Media.

Wood, M. (2014). A new kind of e-commerce adds a personal touch. *New York Times* .

Online Appendix Companion to A Dynamic Clustering Approach to Data-Driven Assortment Personalization

A Proofs

Detailed Derivation of the approximate lower bound $G_{1l}(T, \mathcal{I}')$. We define product j's optimality gap for customers with profile i as $\Delta_j^i := \max_{\{1\le l\le N\}}\{\mu_l^i\} - \mu_j^i$ for $j \in \mathcal{N}$ and $i \in \mathcal{I}$. Let $k_j^i(t)$ denote the number of times that product j has been offered to a customer with profile i up to (and including) time t. One can rewrite the retailer's regret associated with a policy π, defined in (3), as $R^\pi(T, I) = \sum_{i=1}^I R_i^\pi(T)$ where $R_i^\pi(T)$ is the regret associated with profile i and policy π:

$$R_i^\pi(T) := \sum_{j=1}^N \Delta_j^i \, \mathbb{E}_\pi\left[k_j^i(T)\right].$$

The regret is proportional to the number of times that suboptimal assortments are offered to customers over the selling season.

In their seminal work, Lai and Robbins (1985) proved that the regret of any consistent policy in a classic bandit setting grows asymptotically with order at least $\ln(T)$. Formally, for any consistent policy π, we have that

$$\liminf_{T\to\infty} \frac{R^\pi(T)}{\ln(T)} \ge \sum_{j\in\mathcal{N}:\Delta_j>0} \frac{\Delta_j}{D_j}, \tag{A.1}$$

where Δ_j is the optimality gap and D_j is the Kullback-Leibler divergence associated with arm $j \in \mathcal{N}$.[11]

Because all profiles within a cluster have the same distribution of preferences, with some abuse of notation, we let Δ_j^k denote the optimality gap of product j for all profiles in cluster k; that is, $\Delta_j^i = \Delta_j^k$ for all $i \in \mathcal{I}_k$. Similarly, we let $D_j^i = D_j^k$ for all $i \in \mathcal{I}_k$. Building on Lai and Robbins' result in (A.1), we consider the following (asymptotic) lower bound on the regret for the data-intensive policy:

$$L_{d\text{-}int}(T, \mathcal{I}') := \sum_{k=1}^K \left(\sum_{j=2}^N \frac{\Delta_j^k}{D_j^k}\right) \sum_{i\in\mathcal{I}_k} \mathbb{E}\left[\ln(T_i)\right],$$

where T_i is the (random) number of customers from profile i that arrive during the selling season. Hence, $\sum_{i=1}^I T_i = T$ almost surely.[12] Note that $L_{d\text{-}int}$ is the sum of the lower bound on the regret associated with each profile (or cluster). Unlike the setting studied in Lai and Robbins (1985), which considers a homogeneous population of customers, the number of customer arrivals from

[11] The Kullback-Leibler divergence measures the difference between two probability distributions. For the Bernoulli distribution, we have $D_j := \mu_j \ln(\mu_j/\mu^*) + (1-\mu_j)\ln((1-\mu_j)/(1-\mu^*))$, where $\mu^* := \max_{j\in\mathcal{N}}\{\mu_j\}$.

[12] Although the lower bound in (A.1) is attained when $T \to \infty$, one can obtain a similar lower bound on the regret associated with each profile i, $R_i^\pi(T)$, as $T_i \to \infty$. This is because $T_i \to \infty$ almost surely as $T \to \infty$, as long as $p_i > 0$ for each profile $i \in \mathcal{I}$.

each profile (i.e, T_i) is random. This introduces additional complexity in the computation of the lower bound, which we address by using a first-order Taylor expansion of the term $\ln(T_i)$.[13] We therefore approximate $L_{\textit{d-int}}$ by replacing $\ln(T_i)$ by its first-order Taylor expansion around the mean $\mathbb{E}(T_i) = p_i T$. To this end, we have that $\ln(T_i) \approx \ln(p_i T) + (1/(p_i T))(T_i - p_i T)$ and taking expectation on both sides, we have that $\mathbb{E}\left[\ln(T_i)\right] \approx \ln(p_i T)$ as $\mathbb{E}(T_i) = p_i T$. Thus, we approximate the lower bound function $L_{\textit{d-int}}$ with $L^{apx}_{\textit{d-int}}$ where

$$L_{\textit{d-int}}(T, \mathcal{I}') \approx L^{apx}_{\textit{d-int}}(T, \mathcal{I}') := \sum_{k=1}^{K} \left(\sum_{j=2}^{N} \frac{\Delta_j^k}{D_j^k} \right) \sum_{i \in \mathcal{I}_k} \ln(p_i T).$$

Using the results in Auer et al. (2002) for the UCB1 policy, an upper bound on the regret for the semi-oracle policy is given by:

$$U_{\textit{s-orc}}(T, \mathcal{I}') := \sum_{k=1}^{K} \left(\sum_{j=2}^{N} \left(\frac{8}{\Delta_j^k} \right) \mathbb{E}\left[\ln(T'_k)\right] + \left(1 + \frac{\pi^2}{3}\right) \sum_{j=2}^{N} \Delta_j^k \right),$$

where T'_k denotes the (random) number of customers from cluster k. Using a similar first-order Taylor expansion as in $L^{apx}_{\textit{d-int}}$, we approximate the upper bound function $U_{\textit{s-orc}}$ with $U^{apx}_{\textit{s-orc}}$ where

$$U_{\textit{s-orc}}(T, \mathcal{I}') \approx U^{apx}_{\textit{s-orc}}(T, \mathcal{I}') := \sum_{k=1}^{K} \left(\sum_{j=2}^{N} \left(\frac{8}{\Delta_j^k} \right) \ln(P_k T) + \left(1 + \frac{\pi^2}{3}\right) \sum_{j=2}^{N} \Delta_j^k \right),$$

where $P_k := \sum_{i \in I_k} p_i$.

We therefore approximate the lower bound function $L_{\textit{d-int}}(T, \mathcal{I}') - U_{\textit{s-orc}}(T, \mathcal{I}')$ by

$$G_{1l}(T, \mathcal{I}') := L^{apx}_{\textit{d-int}}(T, \mathcal{I}') - U^{apx}_{\textit{s-orc}}(T, \mathcal{I}').$$

Since the gap function can be defined as the gap associated with each cluster, we further define $G_{1l}(T, \mathcal{I}') := \sum_{k=1}^{K} G^k_{1l}(T, \mathcal{I}_k)$ where

$$G^k_{1l}(T, \mathcal{I}_k) := \left[\left(\sum_{j=2}^{N} \frac{\Delta_j^k}{D_j^k} \right) \sum_{i \in \mathcal{I}_k} \ln(p_i T) \right] - \left[\sum_{j=2}^{N} \left(\frac{8}{\Delta_j^k} \right) \ln(P_k T) + \left(1 + \frac{\pi^2}{3}\right) \sum_{j=2}^{N} \Delta_j^k \right].$$

□

The next result provides conditions under which $G_{1l}(T, \mathcal{I}') \geq 0$.

Theorem A.1. *For each $k \in \{1, \ldots, K\}$, suppose that $I_k \geq I_l^k$ for some I_l^k independent of T and of the distribution of customer arrivals. Then, there exist thresholds $t_l^k(I_k)$ such that $G_{1l}(T, \mathcal{I}') \geq 0$ for $T \geq \max_k \left\{ t_l^k(I_k) \right\}$.*

[13] In deriving the bounds, we use a first-order Taylor expansion which, as one may expect, provides a reasonable approximation for smooth functions – we also verified this numerically.

Proof. (i) To prove that $G_{1l}(T,\mathcal{I}') \geq 0$, we first characterize conditions under which $G_{1l}^k(T,\mathcal{I}_k) \geq 0$. After some algebra, we obtain that

$$G_{1l}^k(T,\mathcal{I}_k) = A\ln(T) - B,$$

where $A := I_k \left(\sum_{j=2}^N \Delta_j^k/D_j^k\right) - \sum_{j=2}^N 8/\Delta_j^k$,

$$B := -\left(\sum_{j=2}^N \Delta_j^k/D_j^k\right)\ln\left(\prod_{i\in\mathcal{I}_k} p_i\right) + \left(\sum_{j=2}^N 8/\Delta_j^k\right)\ln(P_k) + \left(1+\pi^2/3\right)\sum_{j=2}^N \Delta_j^k,$$

$P_k = \sum_{i\in\mathcal{I}_k} p_i$, and D_j^k is the Kullback-Leibler divergence number. Let I_l^k be the smallest integer greater than $\left(\sum_{j=2}^N 8/\Delta_j^k\right) / \left(\sum_{j=2}^N \Delta_j^k/D_j^k\right)$ and set $I_k \geq I_l^k$. Note that $A > 0$ for $I_k \geq I_l^k$. We therefore have that $G_{1l}^k(T,\mathcal{I}_k) \geq 0$ if

$$T \geq t_l^k(I_k) := \lceil \exp(B/A) \rceil.$$

The result follows from noting that $G_{1l}(T,\mathcal{I}') = \sum_{k=1}^K G_{1l}^k(T,\mathcal{I}_k)$.
(ii) This part follows immediately as $G_{1l}(T,\mathcal{I}') = \sum_{k=1}^K G_{1l}^k(T,\mathcal{I}_k)$ where $G_{1l}^k(T,\mathcal{I}_k) = A\ln(T) - B$ (where A and B are as defined in part (i)), and $A > 0$ for $I_k \geq I_l^k$. □

Proof of Theorem 1. We assume for simplicity that I_k is continuous. The results follow directly for the discrete case. We prove the results by taking the first- and second-order partial derivatives. We first prove that $\partial G_{1l}(T,\mathcal{I}')/\partial I_k > 0$. We have that $\ln\left(\prod_{i\in\mathcal{I}_k} p_i\right) = I_k\ln(P_k) - I_k\ln(I_k)$. Then,

$$\frac{\partial G_{1l}(T,\mathcal{I}')}{\partial I_k} = \left(\sum_{j=2}^N \frac{\Delta_j^k}{D_j^k}\right)\left(\ln(T) + \ln(P_k) - \ln(I_k) - 1\right).$$

Note that $\partial G_{1l}(T,\mathcal{I}')/\partial I_k > 0$ if $T > eI_k/P_k$. To prove concavity, note that

$$\frac{\partial^2 G_{1l}(T,\mathcal{I}')}{\partial {I_k}^2} = -\left(\sum_{j=2}^N \frac{\Delta_j^k}{D_j^k}\right)\left(\frac{1}{I_k}\right) < 0.$$

□

Detailed Derivation of the approximate lower bound $G_{2l}(T, I, P)$. We assume that $\Delta_j^k = \Delta > 0$ for all $k = 1, \ldots, K$ and $j \neq k$, where recall that Δ_j^k denotes the optimality gap of product j for all profiles in cluster k. Similarly, we let $D_j^k = D > 0$ for all $k = 1, \ldots, K$ and $j \neq k$, where D_j^k denotes the Kullback-Leibler divergence associated with product j for all profiles in cluster k. As in derivation of the approximate lower bound $G_{1l}(T, \mathcal{I}')$, we consider the (asymptotic) lower bound on the regret of the data-intensive policy:

$$L_{d\text{-}int}(T, I, P) := \frac{(K-1)\Delta}{D} \sum_{i=1}^{I} \mathbb{E}\left[\ln(T_i)\right].$$

Existing upper bounds in the bandit literature (including that of the UCB1 policy) assume a homogeneous market. This introduces additional complexity in the computation of the upper bound on the regret of the pooling policy, which we address in the following result.

Lemma 1. *The regret for the pooling policy in this setting is at most*

$$U_{pool}(T, I, P) = \left[\frac{8\ln(T)}{\Delta} \sum_{k=2}^{K} \left(\frac{1}{P_1 - P_k}\right) + \left(1 + \frac{\pi^2}{3}\right)(KP_1 - 1)\Delta\right] + T(1 - P_1)\Delta.$$

Note that U_{pool} in Lemma 1 has an additional term compared to its counterpart in the case of a homogeneous market. The term inside square brackets is the upper bound on the regret of the pooling policy relative to a "weak" oracle. The "weak" oracle assumes that the market is homogeneous and therefore always offers product 1 (as it has a higher purchase probability averaged over all customer profiles) to any arriving customer. However, the regret has to also account for the pooling policy's inability to differentiate between clusters. The second term in U_{pool} incorporates such a penalty. Note that this penalty term disappears as $P_1 \uparrow 1$.

Let $G_{2l}(T, I, P)$ be the approximation of the lower bound $L_{d\text{-}int}(T, I, P) - U_{pool}(T, I, P)$, obtained by replacing $\ln(T_i)$ with its first-order Taylor expansion. That is,

$$G_{2l}(T, I, P) := \left[\frac{(K-1)\Delta}{D} \sum_{i=1}^{I} \ln(p_i T)\right] - \left[\frac{8\ln(T)}{\Delta} \sum_{k=2}^{K} \left(\frac{1}{P_1 - P_k}\right) + T(1 - P_1)\Delta + \left(1 + \frac{\pi^2}{3}\right)(KP_1 - 1)\Delta\right].$$

□

Proof of Lemma 1. We construct U_{pool} in two steps. First, consider a "weak" oracle that assumes that the market is homogeneous, and therefore, presumes that the purchase probabilities are $P_1\mu_j^1 + P_2\mu_j^2 + \cdots + P_K\mu_j^K$ for each product j, where μ_j^k denotes the purchase probability of product j for cluster k. Since the optimality gap of products in all clusters is the same (i.e., $\Delta_j^k = \Delta$ for any cluster k and product $j \neq k$) and $P_1 > \max\{P_2, \ldots, P_K\}$, the weak oracle offers product 1 to all arriving customers regardless of their profiles. As a result, the optimality gap of product $k = 2, \ldots, K$ (averaged over all profiles) for the weak oracle is:

$$\Delta_k' := (P_1\mu_1^1 + P_2\mu_1^2 + \cdots + P_K\mu_1^K) - (P_1\mu_k^1 + P_2\mu_k^2 + \cdots + P_K\mu_k^K) = (P_1 - P_k)\Delta > 0.$$

If the pooling policy follows the UCB1 algorithm of Auer et al. (2002) (which assumes a homogeneous market), its regret relative to the weak oracle is at most

$$\sum_{k=2}^{K}\left[\frac{8\ln(T)}{\Delta'_k}+\left(1+\frac{\pi^2}{3}\right)\Delta'_k\right]=\frac{8\ln(T)}{\Delta}\sum_{k=2}^{K}\left(\frac{1}{P_1-P_k}\right)+\left(1+\frac{\pi^2}{3}\right)(KP_1-1)\Delta.$$

The pooling policy's regret, however, has to also account for its inability to differentiate between the clusters. Now, consider a "strong" oracle that knows the purchase probabilities of all customer profiles (and thus the mapping of profiles to clusters). The strong oracle offers product k to customers from cluster k for $k=1,\ldots,K$, while the weak oracle offers product 1 to all arriving customers regardless of their profiles (or clusters). As a result, the weak oracle incurs an additional cost (regret) relative to the strong oracle whenever a customer from a cluster other than cluster 1 arrives. Hence, the weak oracle's regret relative to the strong oracle due to the lack of knowledge of the mapping of profiles to clusters is $T(1-P_1)\Delta$. Therefore, we obtain

$$U_{pool}(T,I,P)=\left[\frac{8\ln(T)}{\Delta}\sum_{k=2}^{K}\left(\frac{1}{P_1-P_k}\right)+\left(1+\frac{\pi^2}{3}\right)(KP_1-1)\Delta\right]+T(1-P_1)\Delta.$$

□

Proof of Theorem 2. Note that if $P_1=1$, this setting coincides with a homogeneous market and thus the results follow from Theorem A.1. Therefore, we prove the results for $P_1<1$.

We first characterize conditions under which $G_{2l}(t,I,P)\geq 0$. For simplicity we assume that t is continuous, but similar arguments apply to the discrete case. After some algebra, we obtain that

$$G_{2l}(t,I,P)=A'\ln(t)-B't-C',$$

where $A':=(K-1)I\Delta/D-8\beta/\Delta$, $B':=(1-P_1)\Delta$, $C':=(-(K-1)\Delta/D)\ln\left(\prod_{i=1}^{I}p_i\right)+\left(1+\pi^2/3\right)(KP_1-1)\Delta$, $\beta:=\sum_{k=2}^{K}1/\left(P_1-P_k\right)$, and D is the Kullback-Leibler divergence. Note that if $A'\leq 0$, then $G_{2l}(t,I,P)<0$ for all $t\geq 1$ as $B'>0$ and $C'>0$. Thus, we need $A'>0$ which requires that

$$I>\frac{8\beta D}{(K-1)\Delta^2}. \tag{A.2}$$

Having $A'>0$ implies that $G_{2l}(t,I,P)$ is a concave function of t, and this function has a maximum at $t=A'/B'$. Let $G_l^{max}:=\max_{t>0}\{G_{2l}(t,I,P)\}=G_{2l}(A'/B',I,P)$. Note that $\partial G_{2l}(t,I,P)/\partial t\geq(<)\ 0$ for $t\leq(>)\ A'/B'$, and $G_{2l}(t,I,P)\to-\infty$ as $t\to\infty$ (since $B'>0$). Therefore, if $G_l^{max}>0$, we have that $G_{2l}(t,I,P)\geq 0$ for $\tilde{t}_l(I,P)\leq t\leq\tilde{t}_u(I,P)$ for some thresholds $1<\tilde{t}_l(I,P)\leq\tilde{t}_u(I,P)\leq\infty$. In what follows, we provide conditions that guarantee that $G_l^{max}>0$. We have that

$$G_l^{max}=A'\ln(A'/B')-A'-C'.$$

It follows that $G_l^{max} > 0$ if

$$\frac{A'}{B'} > \exp\left(\frac{C'}{A'} + 1\right). \tag{A.3}$$

Considering the fact that arrivals are uniform, i.e., $p_i = 1/I, \forall i$, we have that $\ln\left(\prod_{i=1}^{I} p_i\right) = -I\ln(I)$. This implies that

$$\begin{aligned}\frac{C'}{A'} &= \frac{((K-1)\Delta/D)\, I\ln(I) + \left(1+\pi^2/3\right)(KP_1-1)\Delta}{(K-1)I\Delta/D - 8\beta/\Delta} \\ &= \ln(I) + \frac{(8\beta/\Delta)\ln(I) + \left(1+\pi^2/3\right)(KP_1-1)\Delta}{(K-1)I\Delta/D - 8\beta/\Delta} \\ &= \ln(I) + f(I,P),\end{aligned}$$

where

$$f(I,P) := \frac{(8\beta/\Delta)\ln(I) + \left(1+\pi^2/3\right)(KP_1-1)\Delta}{(K-1)I\Delta/D - 8\beta/\Delta}.$$

Therefore, to ensure that (A.3) holds, we provide conditions that guarantee that

$$\frac{A'}{B'} > \exp(\ln(I) + f(I,P) + 1) = (\exp(f(I,P)+1))\, I. \tag{A.4}$$

After some algebra, we obtain that (A.4) is true if

$$\left(\frac{K-1}{(1-P_1)D} - \exp(f(I,P)+1)\right) I > \frac{8\beta}{(1-P_1)\Delta^2}. \tag{A.5}$$

Note that for the above inequality to hold, the left-hand side of (A.5) must be positive. This requires that $P_1 > \Pi_1(I,P)$ where

$$\Pi_1(I,P) := 1 - \frac{K-1}{D\exp(f(I,P)+1)}.$$

For any given I and P', we let $\tilde{P}_1(I,P') := \inf\{P_1 : P_1 > \Pi_1(I,P), \max\{P_2,\ldots,P_K\} < P_1 < 1\}$ where $P_K = 1 - (P_1 + P_2 + \cdots + P_{K-1})$. Note that $\Pi_1(I,P) < 1$. Moreover, in what follows (Lemma 2), we show that $f(I,P)$ (and thus $\Pi_1(I,P)$) is decreasing in P_1 for $I > e$. Therefore, such $\tilde{P}_1(I,P')$ exists for $I > e$. It follows that, if $P_1 > \tilde{P}_1(I,P')$ and $I > e$, then after some algebra we have that (A.5) is true if

$$I > \frac{8\beta D}{\Delta^2\left((K-1) - \exp(f(I,P)+1)(1-P_1)D\right)}. \tag{A.6}$$

We now present the intermediate result.

Lemma 2. *$f(I,P)$ is decreasing in I and P_1 (assuming that the vector P' is constant) for $I > e$.*

Proof. We prove the result by taking the first-order partial derivatives. We have that

$$\begin{aligned}\frac{\partial f}{\partial I} &= \frac{8(K-1)\beta/D - (8\beta/\Delta)^2(1/I) - ((K-1)\Delta/D)\left[(8\beta/\Delta)\ln(I) + \left(1+\pi^2/3\right)(KP_1-1)\Delta\right]}{((K-1)I\Delta/D - 8\beta/\Delta)^2} \\ &= \frac{[(8(K-1)\beta/D)(1-\ln(I))] - (8\beta/\Delta)^2(1/I) - \left(1+\pi^2/3\right)(K-1)(KP_1-1)\Delta^2/D}{((K-1)I\Delta/D - 8\beta/\Delta)^2}.\end{aligned}$$

Note that $(8(K-1)\beta/D)(1-\ln(I)) < 0$ guarantees that $\partial f/\partial I < 0$. The former inequality holds if $I > e$.

We note that $P_K = 1 - (P_1 + P_2 + \cdots + P_{K-1})$. We then have that

$$\frac{\partial \beta}{\partial P_1} = -\left[\frac{1}{(P_1-P_2)^2} + \frac{1}{(P_1-P_3)^2} + \cdots + \frac{1}{(P_1-P_{K-1})^2} + \frac{2}{(2P_1+P_2+P_3+\cdots+P_{K-1}-1)^2}\right] < 0.$$

Thus, β is decreasing in P_1 which implies that the denominator of function f is increasing in P_1. Therefore, to show that function f is decreasing in P_1, we provide conditions that guarantee that its numerator is decreasing in P_1. Let f_{num} denote the numerator of function f. We then have that

$$\frac{\partial f_{num}}{\partial P_1} = -\left(8\ln(I)/\Delta\right)\gamma + K\left(1+\pi^2/3\right)\Delta,$$

where $\gamma := -\partial\beta/\partial P_1$. Therefore, $\partial f_{num}/\partial P_1 < 0$ if

$$\ln(I) > \frac{K\left(1+\pi^2/3\right)\Delta^2}{8\gamma}.$$

Since $\gamma \geq K$, we have that $I > \exp\left(\left(1+\pi^2/3\right)\Delta^2/8\right)$ guarantees that $\partial f/\partial P_1 < 0$. Because $\exp(1) = e > \exp\left(\left(1+\pi^2/3\right)\Delta^2/8\right)$ as $0 < \Delta \leq 1$, we conclude that $I > e$ guarantees that $\partial f_{num}/\partial P_1 < 0$.

□

We conclude that in order to have $G_l^{max} > 0$, we need that $P_1 > \tilde{P}_1(I, P')$ and the inequalities (A.2) and (A.6) to hold. Seeing that the right-hand side of inequality (A.6) is larger than that for inequality (A.2), for any given P, we set $\tilde{I}_l(P)$ to be the smallest integer I greater than e that satisfies the inequality (A.6) and take $I \geq \tilde{I}_l(P)$. From Lemma 2, we know that $f(I, P)$ is decreasing in I for $I > e$. Because the right-hand side of the inequality (A.6) is increasing in $f(I, P)$ (and thus decreasing in I for $I > e$), such $\tilde{I}_l(P)$ exists (note that the right-hand side of the inequality (A.6) converges to $8\beta D/\left(\Delta^2((K-1) - (1-P_1)eD)\right)$ as $I \to \infty$). This completes the proof of $G_{2l}(t, I, P) \geq 0$.

We next prove that $\tilde{I}_l(P)$ is non-increasing in P_1 (assuming that the vector P' is constant). We know that $\tilde{I}_l(P)$ is the smallest integer I greater than e that satisfies the inequality in (A.6). The result follows from noting that the right-hand side of inequality (A.6) is decreasing in P_1 for $I > e$ as a result of Lemma 2.

We finally prove that $\tilde{P}_1(I, P')$ is non-increasing in I. We have from above that

$$\tilde{P}_1(I, P') = \inf\left\{P_1 : P_1 > \Pi_1(I, P), \max\left\{P_2, \ldots, P_K\right\} < P_1 < 1\right\},$$

where $\Pi_1(I, P) = 1 - (K-1)/(D\exp(f(I, P) + 1))$. The result follows from noting that $\Pi_1(I, P)$ is decreasing in I for $I > e$ as a result of Lemma 2.

□

B Extension of Theoretical Results for Thompson Sampling

Consider a classic bandit setting with N arms and Bernoulli rewards and suppose, without loss of generality, that arm 1 is the unique optimal arm. Let Δ_j denote the optimality gap of product j. Agrawal and Goyal (2011) prove the following upper bound on the regret of Thompson Sampling policy in this setting:

$$R^{\pi}(T) \leq C_1 \ln(T) + C_2,$$

where π denotes the Thompson Sampling policy,

$$C_1 := 1152 \left(\sum_{j=2}^{N} \frac{1}{\Delta_j^2} \right)^2 + 288 \sum_{j=2}^{N} \frac{1}{\Delta_j^2} + 72 \sum_{j=2}^{N} \frac{1}{\Delta_j}$$

and

$$C_2 := 192N \sum_{j=2}^{N} \frac{1}{\Delta_j^2} + 104(N-1).$$

Using the upper bound above, extending the results of Section 6.1 is immediate. In what follows, we discuss the changes to the proof of Lemma 1 and Theorem 2. Following the same arguments as in the proof of Lemma 1, we have in this setting that

$$U_{pool}(T, I, P) = \left(C_1' \ln(T) + C_2' \right) + T(1 - P_1)\Delta,$$

where

$$C_1' := 1152 \left(\sum_{k=2}^{K} \frac{1}{{\Delta_k'}^2} \right)^2 + 288 \sum_{k=2}^{K} \frac{1}{{\Delta_k'}^2} + 72 \sum_{k=2}^{K} \frac{1}{\Delta_k'}$$

and

$$C_2' := 192N \sum_{k=2}^{K} \frac{1}{{\Delta_k'}^2} + 104(N-1),$$

and $\Delta_k' = (P_1 - P_k)\Delta$ is as defined in the proof of Lemma 1. After some algebra, we obtain that

$$G_{2l}(t, I, P) = A' \ln(t) - B't - C',$$

where $A' := (K-1)I\Delta/D - C_1'$, $B' := (1 - P_1)\Delta$, $C' := (-(K-1)\Delta/D) \ln\left(\prod_{i=1}^{I} p_i\right) + C_2'$. Following the same arguments as in the proof of Theorem 2, we need $A' > 0$ which requires that

$$I > \frac{C_1' D}{(K-1)\Delta}. \tag{B.1}$$

The only change to the rest of the proof is the definition of the functions $f(I, P)$ and $\tilde{I}_l(P)$, and inequality (A.6). In this setting, we have that

$$f(I, P) := \frac{C_1' \ln(I) + C_2'}{(K-1)I\Delta/D - C_1'}.$$

After some algebra, we have that the inequality (A.6) in this setting becomes

$$I > \frac{C_1' D}{\Delta\left((K-1) - \exp(f(I,P)+1)(1-P_1)D\right)}. \tag{B.2}$$

Similar to the proof of Theorem 2, we note that the right-hand side of inequality (B.2) is larger than that for inequality (B.1). Thus, we define $\tilde{I}_l(P)$ as the smallest integer I greater than e that satisfies the inequality in (B.2).

To conclude the extension, we only need to prove Lemma 2 in this setting. Following the same arguments as in the proof of Lemma 2, we have that

$$\frac{\partial f}{\partial I} = \frac{\left[\left(C_1'(K-1)\Delta/D\right)(1-\ln(I))\right] - {C_1'}^2/I - \left((K-1)\Delta C_2'/D\right)}{\left((K-1)I\Delta/D - C_1'\right)^2}.$$

Note that $I > e$ guarantees that $\partial f/\partial I < 0$. To show that $f(I,P)$ is decreasing in P_1, note that Δ_k' is increasing in P_1 which, in turn, implies that C_1' and C_2' are decreasing in P_1.

The non-increasing property of $\tilde{I}_l(P)$ and $\tilde{P}_1(I,P')$ in P_1 and I, respectively, follows from similar arguments as in the proof of Theorem 2 and the discussion above.