Management Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

A Dynamic Clustering Approach to Data-Driven Assortment Personalization

Fernando Bernstein, Sajad Modaresi, Denis Sauré

MANAGEMENT SCIENCE
Articles in Advance, pp. 1–21
ISSN 0025-1909 (print), ISSN 1526-5501 (online)
http://pubsonline.informs.org/journal/mnsc/

A Dynamic Clustering Approach to Data-Driven Assortment Personalization

Fernando Bernstein,[a] Sajad Modaresi,[b] Denis Sauré[c]

[a] Fuqua School of Business, Duke University, Durham, North Carolina 27708; [b] Kenan-Flagler Business School, University of North Carolina at Chapel Hill, Chapel Hill, North Carolina 27599; [c] Industrial Engineering Department, University of Chile, Republica 701, Santiago, Chile

Contact: fernando.bernstein@duke.edu, http://orcid.org/0000-0002-1013-6898 (FB); sajad_modaresi@kenan-flagler.unc.edu (SM); dsaure@dii.uchile.cl (DS)

Received: April 18, 2016
Revised: June 7, 2017; September 12, 2017; November 11, 2017
Accepted: December 13, 2017
Published Online in Articles in Advance: September 5, 2018

https://doi.org/10.1287/mnsc.2018.3031

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 data set from a large Chilean retailer. The case study suggests that the benefits of the dynamic clustering policy under the MNL model 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.

History: Accepted by Gad Allon, operations management.
Funding: The research of Dr. Sauré was partially funded by the National Fund for Scientific and Technological Development (FONDECYT) [Project 11140261] and the Complex Engineering Systems Institute (ISCI) [ICM-FIC P05-004-F, CONICYT FB0816].
Supplemental Material: The online appendix is available at https://doi.org/10.1287/mnsc.2018.3031.

Keywords: data-driven assortment planning • personalization • dynamic clustering • multiarmed bandit

1. Introduction

Motivation and Objective. According to a recent study by eMarketer (eMarketer 2017), worldwide business-to-consumer e-commerce sales will reach more than $4 trillion in 2021. With the rapid growth of online sales, retailers are finding unprecedented opportunities to both enhance the customer experience and increase revenue. A customer-centric practice that is becoming increasingly pervasive in online retailing is assortment personalization. This is the practice of offering a product mix that is tailored to each customer's taste based on previously collected data (Arora et al. 2008). The benefits of personalization are twofold: on the 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).

There are numerous examples of personalization in online retailing. 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). Among these start-ups is Stitch Fix, an online personalized clothing service that periodically sends its customers boxes containing five pieces of clothing based on each customer's taste (e.g., size, favorite brand and color, budget). Trunk Club is another personalized clothing service.

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.

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 on 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 that, 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 multiarmed bandit problem (Thompson 1933) 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 of 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 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 estimates the mapping of profiles to clusters as well as 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 clusters 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 predetermined.

We illustrate the practical value of the dynamic clustering policy in a realistic setting by using a data set from a large Chilean retailer. We use the case study to quantify the efficiency and demonstrate the implementability of the dynamic clustering policy. The data set 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, under the MNL model, the proposed policy generates more than 37% additional transactions (on average) compared to the data-intensive policy in the case study. This finding quantifies the potential benefit of leveraging similarity in customer preferences by adaptively pooling information across profiles with similar purchasing 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 an MNL 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 under the MNL demand model. While this can be partially explained by the fact that preferences in the data set do not exhibit a linear structure, the advantage of the dynamic clustering policy persists even when using synthetic data (generated based on the data set) 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, 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. This 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 Online Appendix A. Online 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 (segmentation) and demand learning with dynamic assortment personalization. The paper contributes to three streams of research: dynamic assortment planning with demand learning, personalization, and segmentation methods. Papers in the related literature generally focus on customer segmentation or assortment optimization as independent decisions. This paper considers both decisions jointly in a dynamic setting. In what follows, we review the relevant literature in more detail.

Most of the work in the related literature considers static segmentation and static assortment decisions. These papers generally focus on assortment planning for a homogeneous population of customers. We refer to Kök et al. (2015) for a comprehensive review of the assortment planning literature and industry practices. See also Chen et al. (2015), who consider personalized assortment decisions in an offline setting given customers' contextual information using Logit models. We compare the performance of our policy to a linear-utility MNL model similar to that in Chen et al. (2015).

Most recent work considers static segmentation with dynamic assortment decisions. There are two bodies of work in this stream. The first line of research mostly focuses on dynamic assortment planning for a homogeneous population of customers with unknown demand. These include Caro and Gallien (2007), Ulu et al. (2012), Rusmevichientong et al. (2010), Sauré and Zeevi (2013), and Agrawal et al. (2017). Most of these papers take a multiarmed bandit approach for dynamic assortment optimization. The second line of research considers dynamic assortment personalization for a heterogeneous population of customers. However, in these papers customer segments are known a priori and remain static over time, whereas in our paper, customer segments are dynamically updated over time as more information becomes available. This stream of work includes Bernstein et al. (2015) and Golrezaei et al. (2014)—both papers study dynamic assortment planning problems with limited inventory. Gallego et al. (2016) study a problem of resource allocation with applications to personalized assortment optimization. Jasin and Kumar (2012) and Ciocan and Farias (2014) study network revenue management problems. Closer to our work is the paper by Kallus and Udell (2016), which studies a high-dimensional dynamic assortment personalization problem with a fixed, but large, number of customer types. The paper uses a nuclear-norm regularized maximum likelihood approach (i.e., a frequentist approach) for estimation, while we use dynamic Bayesian estimation that updates the number of customer segments based on the observed transaction data.

In contrast to the existing literature, our paper considers simultaneous dynamic customer segmentation and dynamic assortment personalization decisions. Wedel and Kamakura (2012) provide a comprehensive review of market segmentation methodologies such as clustering, mixture models, and profiling segments. Our 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 that 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 customer population. For example, Ansari and Mela (2003) use this model to customize the design and content of emails to increase web traffic (click-through). 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.

From a methodological standpoint, the dynamic assortment selection problem we study can be thought of as a multiarmed 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 multiarmed bandit problem with multiple plays. The dynamic assortment selection problem can alternatively be formulated as a combinatorial multiarmed 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 multiarmed 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 multiarmed 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, 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. 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 $x = (x_{\text{gender}}, x_{\text{age}}, x_{\text{location}})$. 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. On 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 $\mu_0^i = 0$ 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 $\mu^i := (\mu_1^i, \ldots, \mu_N^i)$, 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

$$\Pi_j(S, \mu^i) := \begin{cases} \dfrac{\nu_j^i}{1 + \sum_{j' \in S} \nu_{j'}^i}, & j \in S \cup \{0\}, \\ 0, & \text{otherwise}, \end{cases} \tag{1}$$

where $\nu_j^i := \exp(\mu_j^i)$ 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, $Z_{j,t}^i = 1$ with probability $\Pi_j(S_t, \mu^i)$. Moreover, we let $Z_{0,t}^i = 1$ if a customer with profile i arriving at time t opts not to purchase any product, and $Z_{0,t}^i = 0$ otherwise.

Independent Demand. In this setting, we assume that the purchasing decision $Z_{j,t}^i$ 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_j^i, independent of the assortment in which it is offered. Thus, in this setting, the vector $\mu^i = (\mu_1^i, \ldots, \mu_N^i)$ 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.,

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

A setting in which this demand model may be appropriate is one in which customers make purchasing decisions across various products that are not necessarily substitutes, as may be in the case of Stitch Fix 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 $Z_t := (Z_{1,t}^{i_t}, \ldots, Z_{N,t}^{i_t})$. Let $\mathcal{F}_t := \sigma((S_\tau, W_\tau), 1 \le \tau \le t), t = 1, \ldots, T$, denote the filtration (history) associated with the assortment and purchasing decisions up to (and including) time t, with $\mathcal{F}_0 = \sigma(\emptyset)$. An admissible assortment selection policy π is a mapping from the available history to assortment decisions such that $S_t \in \mathcal{S}$ is nonanticipating (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:

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

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 $K \le I$ 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 expedite the estimation of preferences. This, in turn, translates into higher revenue for the retailer.

Connection to the Multiarmed 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 multiarmed bandit problem is the standard framework for addressing this trade-off. The assortment selection problem can be formulated as a multiarmed bandit by means 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 multiarmed 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,

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

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

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

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 that 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).

4. Dynamic Assortment Personalization

In this section, we introduce a prescriptive approach for dynamic assortment personalization that 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 $F(\cdot \mid \mu^i)$ 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, $F(Z^i_t = e^j \mid \mu^i) = \Pi_j(S, \mu^i)$, where $\Pi_j(S, \mu^i)$ is as defined in (1) and e^j denotes the jth unit vector (although $F(\cdot \mid \mu^i)$ also depends on the assortment S, we drop such dependence to simplify notation). For the case of independent demand, $F(Z^i_{j,t} = 1 \mid \mu^i) = \Pi_j(S, \mu^i)$, 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 $F(\cdot \mid \mu^i)$. 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:

$$Z_t^i \mid \mu^i \sim F(\cdot \mid \mu^i), \tag{4a}$$
$$\mu^i \mid H \sim H, \tag{4b}$$
$$H \sim \mathrm{DP}(H_0, \alpha). \tag{4c}$$

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, 1983) and Neal (2000).

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 $S_t \in \mathscr{S}$ to offer customer t based on the current preference and mapping estimates. (For the first customer arrival, start with an arbitrary mapping from 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.

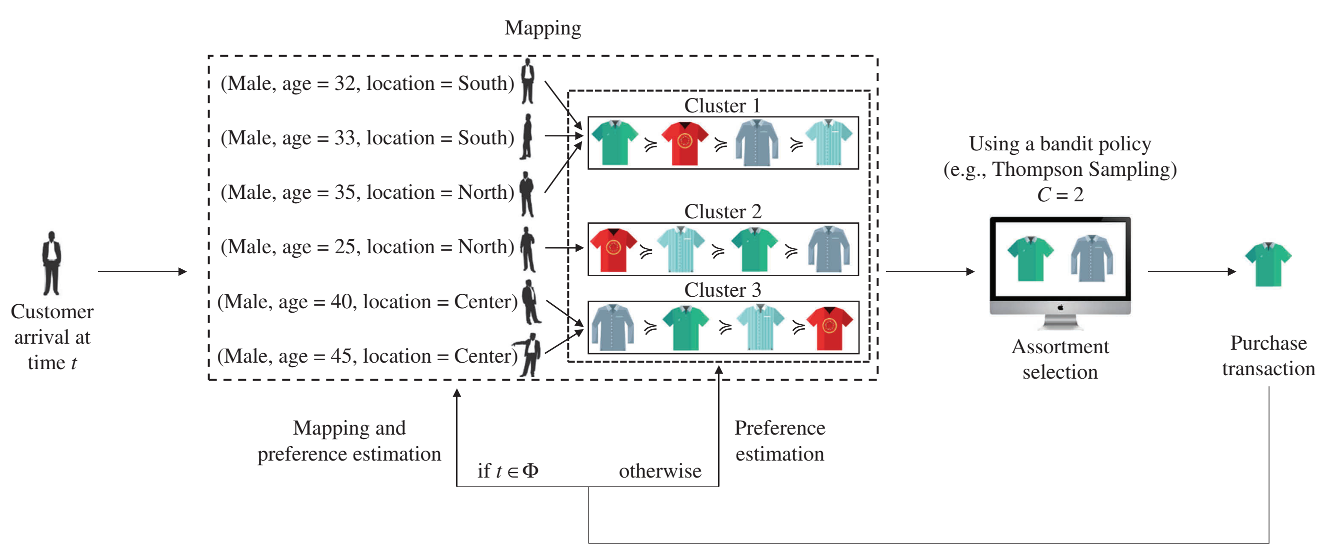
Step 4 (*Mapping and Preference Estimation*). If $t \in \Phi$, then perform the estimation procedure described in Section 4.3 to estimate preferences and mapping of profiles to clusters using all available information; otherwise, only update the preference estimates using the prevailing mapping estimate.

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

Figure 1. (Color online) Illustration of the Dynamic Clustering Policy



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 $\mathscr{X}_t^i$ denote the assortment and purchase history associated with profile i up to (and including) time $t \le T$. That is, $\mathscr{X}_t^i := \{(S_u, Z_u^{i_u})\colon i_u = i, 1 \le u \le t\}$, and set $\mathscr{X}_0^i = \varnothing$. Define $\mathscr{X}_t := (\mathscr{X}_t^1, \dots, \mathscr{X}_t^I)$. 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 from profiles to clusters and sample the preference parameters from H_0 for each cluster. For $s = 1, \dots, \eta$, 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 $i \in \mathscr{I}$, 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]

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

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

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

and do not change c_i with probability $1 - a(c_i^*, c_i)$, where $L(\mathscr{X}_t^i, \mu^{c_i})$ denotes the likelihood function given the purchase history $\mathscr{X}_t^i$ 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 $\mathscr{X}_t$) 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 ("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 $m \le m'$ as the mappings corresponding to several sample points may be identical. Let $0 < f_1, f_2, \dots, f_m \le 1$ be the associated frequency proportions (i.e., the relative number of occurrences of each mapping in the set of selected samples), with $\sum_{n=1}^m f_n = 1$. We approximate the posterior distribution of the mapping as a discrete probability distribution that takes the value M_n with probability f_n, $n \le m$. 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 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 $t-1$ 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

$$\hat{\nu}_{j,t}^{M_s(i)} := \frac{\sum_{l=1}^{t-1} Z_{j,l}^{i_l} \mathbf{1}\{j \in S_l, M_s(i_l) = M_s(i)\}}{\sum_{l=1}^{t-1} Z_{0,l}^{i_l} \mathbf{1}\{j \in S_l, M_s(i_l) = M_s(i)\}}, \quad j \in \mathscr{N}. \quad (5)$$

We then estimate $\mu_j^{M_s(i)}$ for cluster $M_s(i)$ 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 undertested 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 v_j^i by $\hat{v}_{j,t}^i$, where

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

and $\hat{v}_{j,t}^{M_l(i)}$ is as defined in (5). 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 $\hat{v}_{j,t}^i$ 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 Beta 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 $\mathrm{Beta}(a_{j,t-1}, b_{j,t-1})$, and offer product $S_t \in \arg\max_{j \in \mathcal{N}} \{Q_{j,t}\}$ 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 data set from a large Chilean retailer. We first provide a brief overview of the data set 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 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. Data Set

The data set that we use for the case study is from a chain of department stores owned by a Chilean multinational company headquartered in Santiago, Chile.

Figure 2. (Color online) Example of an Assortment Shown on the Retailer's Website

The company sells clothing, footwear, 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 data set 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 data set consists of 94,622 customer-tied click records for a set of 19 products in the footwear category (see Figure 2).[6] The data set used in this study was collected through an experiment in which the 19 products were randomly assigned to eight different assortments of four 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 eight 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 data set. Otherwise, a no-click was recorded. Therefore, each customer visit resulted in at most one click record. The data set 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 seven 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 data set in fact contains more granular information on the age of customers, but we mostly use the age groups as determined by the retailer.)

5.2. Implementation Details

Estimation of Underlying Demand Model. We begin by using the data set 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 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 data set. 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 v_j^i by

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

Note that, as in (5), 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 faster learning 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 data set available from the 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

$$\mu_j^x := \beta_j^\top x = \beta_j^0 + \beta_j^M x_M + \beta_j^{A_2} x_{A_2} + \beta_j^{A_3} x_{A_3} + \beta_j^{L_2} x_{L_2} + \beta_j^{L_3} x_{L_3} + \cdots + \beta_j^{L_7} x_{L_7}, \quad (6)$$

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. 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 $\eta_d = 10$th MCMC draw is used to estimate the mapping of profiles to clusters (this alleviates the autocorrelation 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 i7-6700 3.40-GHz CPU and 16 GB 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 data set from the Chilean retailer. 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.

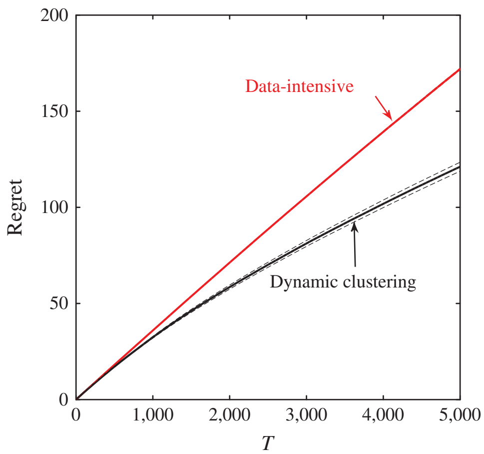
The underlying demand model and distribution of customer arrivals are estimated from the data set. We run the experiments in markets with $T = 5{,}000$ customers. There are 19 products ($N = 19$), and the display constraint is of size four ($C = 4$). Moreover, there are $I = 42$ distinct customer profiles. 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.

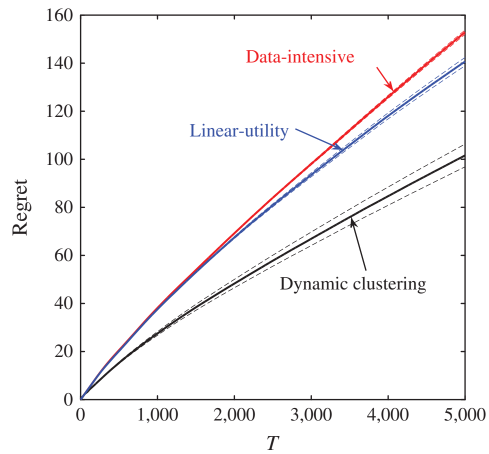
Figure 3 illustrates the average performance for the case of independent demand (left panel) and MNL demand (right panel). The dynamic clustering policy significantly outperforms the data-intensive and linear-utility policies as a result of pooling information. At the same time, the linear-utility policy 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., over customer arrivals). 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 offerings using a larger number of clusters.

In sum, 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

Figure 3. (Color online) Average Performance in a Market with $I = 42$ Profiles and Independent Demand (Left), and MNL Demand (Right)





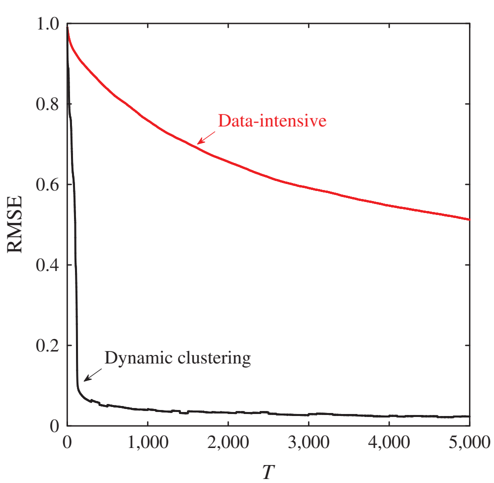
on the Chilean retailer's data set. We find that, under the MNL model, the dynamic clustering policy results (on average) in 37.7% and 27.3% additional transactions compared to the data-intensive and linear-utility policies, respectively. Moreover, the dynamic clustering policy results in more than 65% 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 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. To study the impact of a finer definition

Figure 4. (Color online) 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 $I = 42$



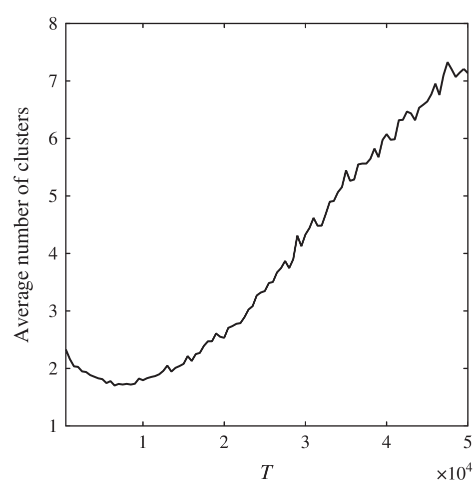
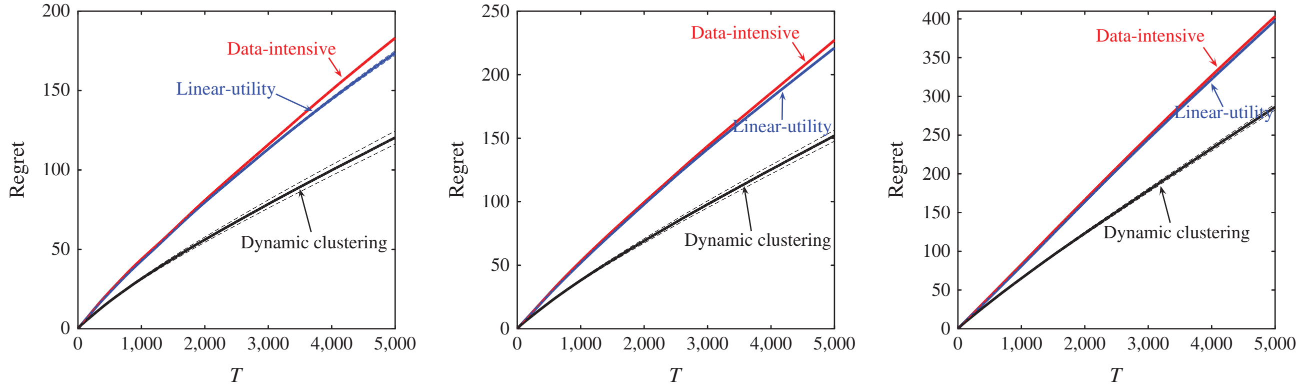


Figure 5. (Color online) Average Performance in a Market with MNL Demand and $I = 75$ (Left), $I = 140$ (Center), and $I = 450$ (Right)



of customer attributes, we have extended the study beyond the 42 original customer profiles used by the retailer. The raw data set available to us contains more granular information 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 $[0, 30], [31, 40], [41, 50], \ldots, [91, 99]$. The second one has 140 customer profiles using age groups $[0, 20], [21, 25], [26, 30], \ldots, [96, 99]$. The third set of experiments has 450 customer profiles using exact ages $18, 19, 20, \ldots, 99$. In each experiment, we only kept the profiles for which there was at least one no-purchase transaction in the data set.

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

Table 1. Average Computation Time (in Seconds) of Different Policies for MNL Demand and $T = 5{,}000$

	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

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.

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 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.

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

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 $T = 5{,}000$

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

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.

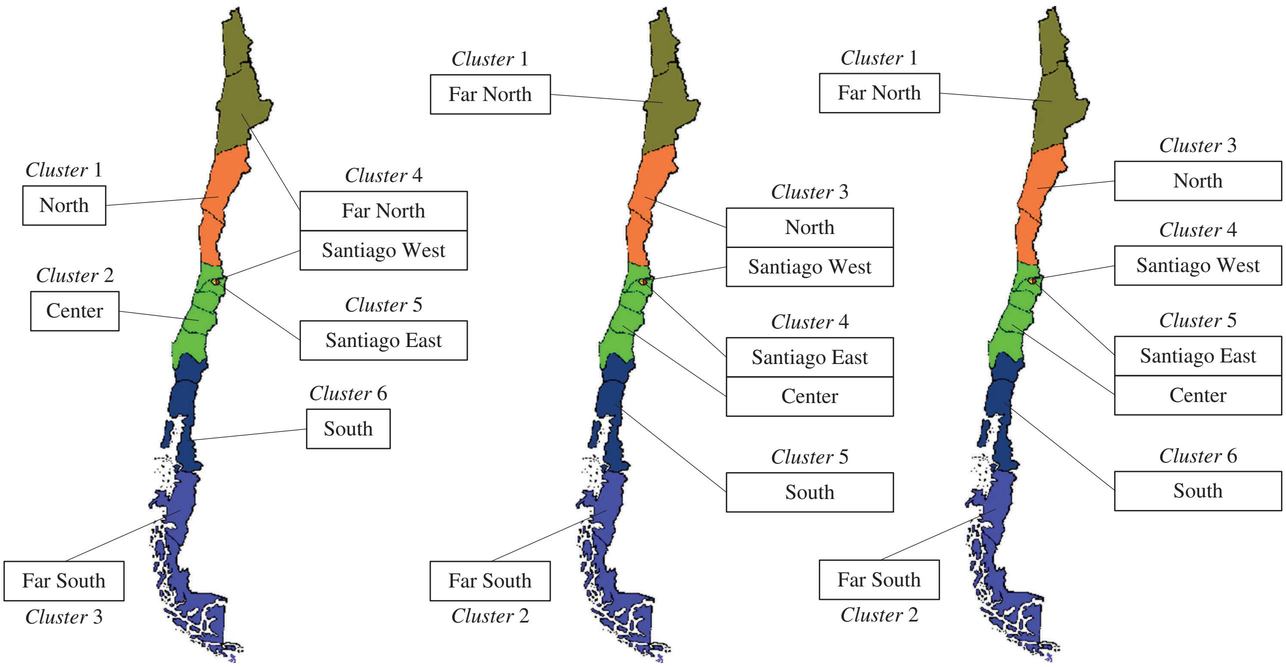
In sum, we find that the better performance of the dynamic clustering policy relative to the other policies is robust with respect to the definition of customer attributes. At the same time, a finer definition of customer attributes (i.e., a larger number of customer profiles) increases the computation 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 for a representative example based on the data set 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 data set from the Chilean retailer exhibits a nonlinear dependence between mean utilities and attributes. In such cases, the linearity assumption could result in inaccurate estimates that, in turn, could hurt the retailer's revenue. Also, the dynamic clustering policy takes a Bayesian 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 data set under the MNL demand model. The linear-utility approach, however, uses maximum-likelihood

Figure 6. (Color online) Illustration of Clusters (Under the Most Likely Mapping) for Women from Age Groups [0, 29] (Left), [30, 39] (Center), and [40, 99] (Right) Based on Customers' Location in Chile



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.

Because the linear-utility approach may be better suited for an offline setting, we designed an additional set of experiments that mimic an offline setting. More specifically, because the estimation approach (and

Figure 7. (Color online) Illustration of Optimal Assortments for Women from Age Group [30, 39]

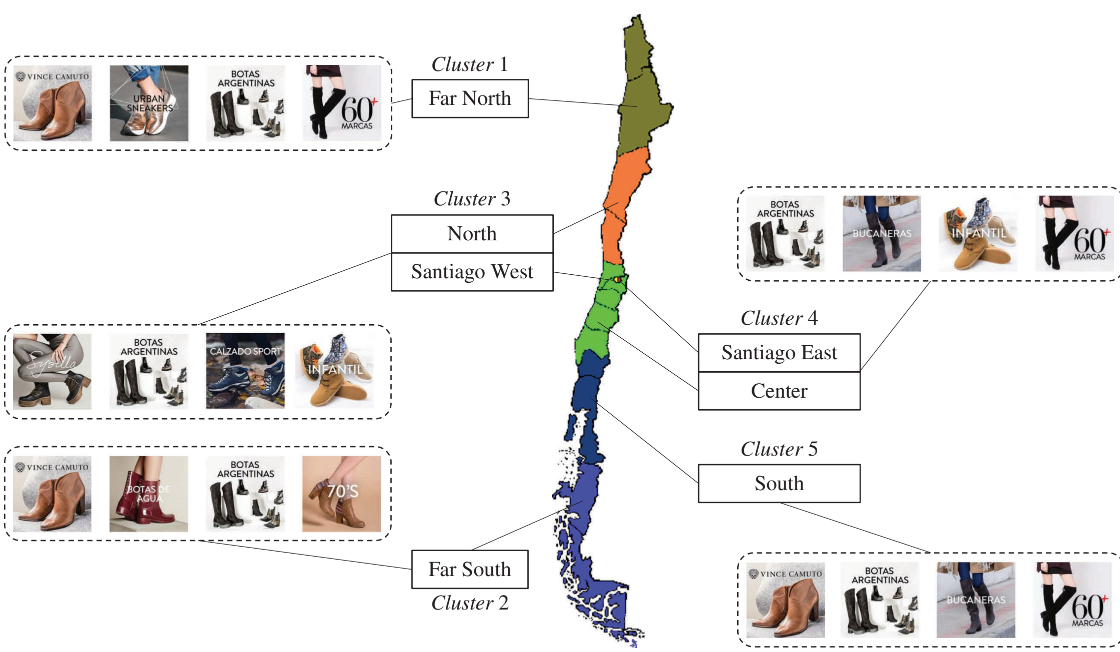


Table 3. Average Percentage Improvement of Dynamic Clustering Policy Over Data-Intensive and Linear-Utility Policies in Separation-Based Experiments Using the Original Data Set

	Regret (%)		Revenue (%)	
	Data-intensive	Linear-utility	Data-intensive	Linear-utility
$T = 500$	34.2	24.6	47.8	25.4
$T = 1{,}000$	42.6	28.4	50.0	21.6
$T = 5{,}000$	39.0	26.6	20.7	10.7
$T = 10{,}000$	24.3	12.9	8.9	3.9
$T = 20{,}000$	6.7	3.6	1.5	0.8

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 that, 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 data set. 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 underlying demand model.

The first scenario is based on the original transaction data set, which exhibits a nonlinear 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 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 data set 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 (6). To that end, we estimate β_j in model (6) for each product j from the data set 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 for that with the largest transaction sample (i.e., $T = 20{,}000$)—please refer to Table B.1 in Online Appendix B for a summary of the results of this numerical study. 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.

Table 4. Average Estimation Time (in Seconds) in Separation-Based Experiments Based on Original Data Set

	$T = 500$	$T = 1{,}000$	$T = 5{,}000$	$T = 10{,}000$	$T = 20{,}000$
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

To conclude, the dynamic clustering policy outperforms the linear-utility approach in the case study based on the Chilean retailer's data set. Each approach has its advantages and may be deemed more appropriate depending on the specific setting. For example, the retailer that provided the data set 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. 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 $R^{\pi^{\text{pool}}}$ 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

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

Our goal is to determine conditions under which these gaps are nonnegative. 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 $L_{d\text{-int}} - U_{\text{pool}}$ provides a lower bound for the gap function G_2. Hence, we focus on characterizing settings in which these lower bounds are nonnegative, 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 bounds on the regrets 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 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 that is simple to use. We extend the results of Sections 6.1 and 6.2 for Thompson Sampling (Agrawal and Goyal 2012) in Online 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 in the dynamic clustering policy) 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 Online Appendix A.)

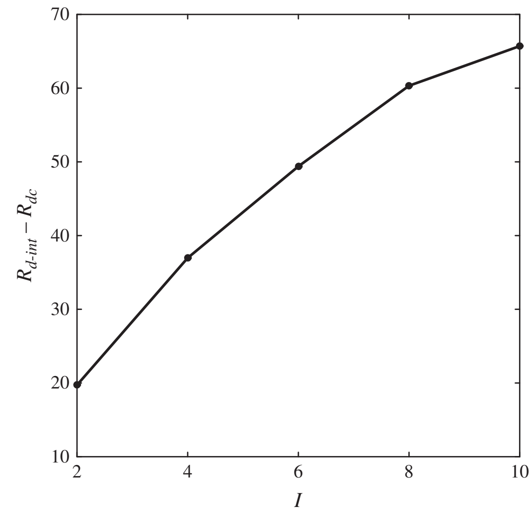
We next show that there are diminishing marginal returns to pooling information from an increasing number of customer profiles. To that end, consider a general market with K clusters where $1 \leq K < I$. We assume, without loss of generality, that $K < I$, since if $K = I$, then both the semi-oracle 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., $\mu^i_j < \mu^i_1$ for $j = 2, \ldots, N$ and all $i \in \mathcal{I}$. Computing the lower bound $L_{d\text{-int}} - U_{s\text{-}orc}$ 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 $L_{d\text{-int}} - U_{s\text{-}orc}$. 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 $\mathcal{I}'$ that encodes the mapping of profiles to clusters. We provide a detailed derivation of $G_{1l}(T, \mathcal{I}')$ in Online 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 data set from the Chilean retailer. Specifically, we estimate the MNL parameters for the customer profile (female, [40, 99], Center) from the data set and assume that a cluster's demand follows such MNL model. We then increase the number of profiles in that cluster

Figure 8. Gap Between the Regrets of Data-Intensive and Dynamic Clustering Policies as a Function of Number of Profiles



and evaluate the gap between the regrets of the data-intensive and dynamic clustering policies in a market with $T = 5{,}000$ 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.

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 $k = 1, \ldots, K$. Furthermore, we assume that $\mu^k_k - \mu^k_j = \Delta$ for some $\Delta > 0$ and for all $k = 1, \ldots, K$ and $j \neq k$, where, to simplify notation, μ^k_j denotes the purchase probability of product j for all profiles in cluster k. Let $P := (P_1, P_2, \ldots, P_K)$ 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 Online Appendix A. Let $G_{2l}(t, I, P)$ be the approximation of the lower bound $L_{d\text{-int}} - U_{\text{pool}}$ to the gap function G_2 at any time period t. We provide a detailed derivation of $G_{2l}(t, I, P)$ in Online 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*

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

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 nonincreasing in P_1 and I, respectively.

The result in Theorem 2 shows that, under some conditions, the pooling policy tends to outperform the data-intensive policy 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 data set 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 heterogeneous. 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. Because the proposed approach relies on MCMC for estimation, the computational cost is a potential concern for implementing the policy in a high-dimensional setting in real-time; nonetheless, the paper discusses several approaches to expedite the computation. Furthermore, the proposed approach does not assume any particular relation between the products' mean utilities and customer attributes. In a high-dimensional setting, imposing further structure on the utility model (e.g., similar to the linear-utility approach discussed in the paper) is a potential alternative to expedite the estimation.

Acknowledgments

The authors are grateful to the department editor, associate editor, and the referees for constructive comments that helped improve the paper.

Endnotes

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

[2] 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.

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

[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.

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

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

[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 nonnegativity 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), $R^{\pi}(T)/T^a \to 0$, as $T \to \infty$, for every $a > 0$. That is, if $R^{\pi}(T) = o(T^a)$. See Lai and Robbins (1985).

References

Agrawal S, Goyal N (2012) Analysis of Thompson sampling for the multi-armed bandit problem. *JMLR: Proc. 25th Conf. Learn. Theory* 23:39.1–39.26.

Agrawal S, Avadhanula V, Goyal V, Zeevi A (2017) MNL-bandit: A dynamic learning approach to assortment selection. Working paper, Columbia University, New York.

Anantharam V, Varaiya P, Walrand J (1987) Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays—Part I: IID rewards. *Automatic Control IEEE Trans.* 32(11):968–976.

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

Antoniak CE (1974) Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *Ann. Statist.* 2(6): 1152–1174.

Arora N, Dreze X, Ghose A, Hess JD, Iyengar R, Jing B, Joshi Y, 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, Fischer P (2002) Finite-time analysis of the multiarmed bandit problem. *Machine Learn.* 47(2–3):235–256.

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

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

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

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

Chen X, Owen Z, Pixton C, Simchi-Levi D (2015) A statistical learning approach to personalization in revenue management. Working paper, New York University, New York.

Ciocan DF, Farias VF (2014) Fast demand learning for display advertising revenue management. Working paper, Massachusetts Institute of Technology Cambridge, MA.

eMarketer (2017) Worldwide retail and ecommerce sales: eMarketer's estimates for 2016–2021. *eMarketer* (July 18), https://www.emarketer.com/Report/Worldwide-Retail-Ecommerce-Sales-eMarketers-Estimates-20162021/2002090.

Ferguson T (1973) A Bayesian analysis of some nonparametric problems. *Ann. Statist.* 1(2):209–230.

Ferguson TS (1983) Bayesian density estimation by mixtures of normal distributions. *Recent Adv. Statist.* 24:287–302.

Fisher M, 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, Wang X (2016) Online personalized resource allocation with customer choice. Working Paper, Columbia University, New York.

Gelman A, Carlin JB, Stern HS, Rubin DB (2014) *Bayesian Data Analysis* (Taylor & Francis, Boca Raton, FL).

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

Heller KA, Ghahramani Z (2005) Bayesian hierarchical clustering. *Proc. 22nd Internat. Conf. Machine Learn.* (ACM), 297–304.

Jasin S, Kumar S (2012) A re-solving heuristic with bounded revenue loss for network revenue management with customer choice. *Math. Oper. Res.* 37(2):313–345.

Kallus N, Udell M (2016) Dynamic assortment personalization in high dimensions. Working paper, Cornell University, Ithaca, NY.

Kök AG, Fisher ML, Vaidyanathan R (2015) Assortment planning: Review of literature and industry practice. Agrawal N, Smith S, eds. *Retail Supply Chain Management*, International Series in Operations Research and Management Science, Vol. 223 (Springer Science & Business Media, New York), 175–236.

Lai TL, Robbins H (1985) Asymptotically efficient adaptive allocation rules. *Adv. Appl. Math.* 6(1):4–22.

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

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

Neal R (2000) Markov chain sampling methods for dirichlet process mixture models. *J. Computational Graphical Statist.* 9(2):249–265.

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

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

Thompson WR (1933) On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika* 25(3–4):285–294.

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

Wedel M, Kamakura WA (2012) *Market Segmentation: Conceptual and Methodological Foundations*, Vol. 8 (Springer Science & Business Media, New York).

Wood M (2014) A new kind of e-commerce adds a personal touch. *New York Times* (August 13), https://www.nytimes.com/2014/08/14/technology/personaltech/data-driven-shopping-with-the-personal-touch.html.