A Nonparametric
Joint Assortment and Price Choice Model

Srikanth Jagabathula
Stern School of Business, New York University, New York, NY 10012, sjagabat@stern.nyu.edu

Paat Rusmevichientong
Marshall School of Business, University of Southern California, Los Angeles, CA 90089, rusmevic@marshall.usc.edu

The selection of products and prices offered by a firm significantly impacts its profits. Existing approaches do not provide flexible models that capture the joint effect of assortment and price. We propose a nonparametric framework in which each customer is represented by a particular price threshold and a particular preference list over the alternatives. The customers follow a two-stage choice process; they consider the set of products with prices less than the threshold and choose the most preferred product from the set considered. We develop a tractable nonparametric expectation-maximization (EM) algorithm to fit the model to the aggregate transaction data and design an efficient algorithm to determine the profit-maximizing combination of offer set and price. We also identify classes of pricing structures of increasing complexity, which determine the computational complexity of the estimation and decision problems. Our pricing structures are naturally expressed as business constraints, allowing a manager to trade off pricing flexibility with computational burden.

Key words: nonparametric choice models, joint assortment and price optimization, EM algorithm, transaction data

---

1. Introduction

What products to carry and what prices to charge are important decisions faced by a firm. These decisions influence the purchase behavior of customers, and, therefore, the firm's revenues and gross profits. To effectively optimize its offerings, a firm must understand how its product and price offerings *jointly* impact consumer demand. The existing literature in the areas of marketing and operations proposes the following general procedure: (i) *model* the impact of assortment and price on consumer demand, (ii) *fit* the model to data on product availability and sales transactions, and (iii) *optimize* the assortment or price or the joint assortment and price offering using the demand predictions from the fitted model. The biggest challenge in executing the above procedure is to design a model that faithfully captures the underlying choice patterns *and* allows for tractable algorithms to fit and optimize.

Most existing proposals are based on parametric choice models, which specify particular functional forms relating product attributes (such as price) to utility values and choice probabilities.

Parametric models are parsimonious and therefore computationally tractable. However, the choice structures must be pre-specified, increasing the risk of model misspecification and leading to inaccuracies in decision making. The alternative is to adopt a nonparametric approach, which removes the need for explicit specification of choice structures and instead 'learns' the appropriate structure from data[1]. However, the lack of parsimonious structures makes model estimation, and *particularly* optimization, computationally difficult. Existing work (e.g., Farias and Jagabathula [15], Haensel and Koole [18], and van Ryzin and Vulcano [38]) proposes computationally efficient techniques to estimate model parameters when data consist of only assortment changes and not price changes. The literature is, however, silent on capturing the joint assortment and price changes and solving the joint assortment and price decision problem using nonparametric approaches.

The key contribution of this paper is a nonparametric approach for joint assortment and price optimization. Despite its flexibility, our model allows for both tractable estimation *and* optimization. Our framework allows managers to make a continuous trade-off between decision complexity and computational burden. We follow the general *model*, *fit*, and *optimize* procedure described above. We focus on a canonical retailer selling a universe of n products from a specific category or sub-category. There are frequent changes in prices and more frequent changes in the offer sets, either due to stock-outs or deliberate screening, which is common in the online environment. Product features, other than prices, remain fixed. The retailer has collected historical data in the form of sales transactions, product availabilities, and offered prices. The retailer must utilize available data to determine the assortment and price combination that maximizes the expected revenue. The above setting is broad and includes the classical revenue management setting for airlines, hotels, and cruises, in which the available bookings and prices change frequently.

Overview of our approach. *Model.* We extend the prevailing nonparametric rank-based choice model to capture the impact of price changes on demand. In a rank-based choice model, customers make choices from an offer set according to a preference list so that if the most preferred option is unavailable, they go down the list to pick an available option, as long as it is preferred over the no-purchase option. We extend the rank-based model by supposing that customers follow a two-stage procedure. In the first stage, the customer forms a *consideration set* by selecting the subset of products whose prices are less than or equal to a *price threshold*. In the second stage, she chooses the most preferred product from the chosen consideration set. Existing literature allows the customer to form a consideration set by applying broader threshold-based screening rules

[1] Indeed, as businesses increasingly move online, such data-driven approaches are essential for businesses to take advantage of the dynamic environments, complex demand patterns, and associated wealth of data offered by online environments.

comprising of attributes other than price, but we focus on the price-based threshold because in our setting only prices change. Of course, neither the ranked list nor the price threshold of the individual customer is observed, so we describe them using probability mass functions (PMFs): the population is described by (a) the *threshold PMF* that describes a distribution over all possible price thresholds, and (b) the *preference PMF* that describes a distribution over all possible ranked lists.

Because customer preferences are influenced by product prices, the preference PMF will be a function of price. But allowing the preference PMF to depend arbitrarily on price results in an intractable model that cannot extrapolate demand to new prices. To address this, we suppose that the domain of price vectors partitions into a "small" number K of partitions such that the preference PMF λ_k is the same for all the price vectors in partition k. In effect, this assumption supposes that price thresholds capture the immediate effects of price changes, while the ranked lists capture the residual effects. The assumption is also supported by empirical evidence from Gilbride and Allenby [17]. It results in tractable estimation because only a finite number of threshold and preference PMFs need to be estimated from data.

Fit. We train the model parameters on historical data that consist of sales transactions, product availabilities, and offered prices. The estimation consists of two steps: (a) clustering the price vectors to result in a partitioning of the price domain and (b) running the expectation-maximization (EM) algorithm to estimate the threshold and preference PMFs for each partition separately. We use a general-purpose clustering algorithm to partition the training price vectors into K parts. We tune K using cross-validation. Given the partitioning, we fit a model to each partition by maximizing the log-likelihood function. Because the log-likelihood function is in general hard to maximize, we adopt the EM algorithm. The EM algorithm treats the price thresholds of individual observations as the latent variables and iteratively alternates between inferring the latent variables (from observed data and previous PMFs) and generating new PMF estimates from the inferred latent variables. The sequence of estimates produced can be shown to converge to a stationary point of the log-likelihood function. The key challenge in running the EM algorithm is carrying out the resulting M-step, which involves solving a convex program with $n!$ (n factorial) number of variables. Practically, we carry out the M-step by reducing it to a rank-aggregation problem and using existing heuristics (see Ali and Meliă [3]) to obtain solutions with good empirical performance. Theoretically, we propose a dynamic programming (DP) formulation to show that the solution can be obtained efficiently (in polynomial time) if we restrict the feasible prices to a structured class.

Optimize. Finally, we optimize the estimated model to determine the revenue-maximizing combination of assortment and prices. The decision problem is NP-hard in the strong sense (Rusmevichientong et al. [30] and Aggarwal et al. [1]). We address this challenge by proposing an approximation algorithm with a provable performance guarantee based on a DP formulation. Under appropriate technical conditions, we show that the DP admits a polynomial-time approximation scheme (PTAS). For any pre-specified $\epsilon > 0$, the PTAS can determine a price vector whose revenue is within $1-\epsilon$ of the optimal revenue, using computing time that is polynomial in the number of products, for a fixed ϵ.

We analyze our method both theoretically and empirically. Theoretically, we identify pricing structures that lend tractability to both the estimation and the optimization problems. We call these the d-sorted pricing structures, the simplest of which ($d = 0$) results in product prices that respect a pre-specified reference ordering[2]. These structures have received little attention in the literature. They allow us to isolate the source of complexity in solving the estimation and optimization problems. Further, they are *not* esoteric mathematical structures but naturally map to a firm's business constraints. Empirically, we test our methods both on real-world and synthetic sales transactions data. The tests with the real-world transactions from the IRI Academic Dataset demonstrate the *predictive* accuracy of our methods: an average of 26% improvement over the benchmark latent-class multinomial logit (LC-MNL) model on a 'chi-square' metric, which measures the relative error in predicting market shares. The tests with the synthetic data demonstrate the *decision* accuracy: an average of 11% higher revenue extracted against the LC-MNL benchmark.

1.1. Literature review

We position our work as part of the new stream of literature on choice modeling techniques designed specifically for applications in operations; see, for example, Farias and Jagabathula [15], Blanchet et al. [8], van Ryzin and Vulcano [38], and Alptekinoglu and Semple [4]. This body of work is characterized by its emphasis on the prediction accuracy rather than on the explanatory power of the models, because accurate decision making requires accurate predictions rather than accurately modeling the underlying choice process. Indeed, the objectives of producing accurate predictions and accurate modeling of underlying choice process are not equivalent (see Ebbes et al. [14]). Furthermore, our work may be viewed as a step toward extending the framework of discrete models of choice in operations to account for behavioral heuristics, such as simple screening rules (see Ben-Akiva et al. [7]). Our work has connections to the literature on choice modeling in both operations and marketing. Because our model is designed for operational decision making, we focus predominantly on the work in operations, with a brief discussion on connections with marketing at the end.

[2] Such sorted pricing structures are generally reasonable when products are vertically differentiated by brand, size, quality, etc.

In operations, rank-based choice models have been used to model demand in the context of airline revenue management and retail operations. Traditional approaches had assumed independent demands for each product. However, if products are close substitutes and their availability changes over time, the demand for each product becomes a function of the entire offer set. As a result, interest has shifted over the last two decades from independent demand models to choice-based demand models. In airline revenue management, the work by Belobaba and Hopperstad [5], Ratliff et al. [29], and Vulcano et al. [39] has demonstrated improvements in revenues from the incorporation of choice models. Retail applications of rank-based choice models have been pioneered by Mahajan and van Ryzin [26], who considered a single-period stochastic inventory model in which customers substitute products within the assortment as inventory is depleted. The above approaches mostly focus on assortment effects on demand.

On the other hand, the research that accounts for price effects on demand has mostly focused on optimization for a given parametric choice model. Hanson and Martin [19] showed that the profit function under the logit choice model is not jointly concave in the price vector. However, there is a one-to-one correspondence between the prices and the market shares, and it has been shown that the profit function is concave in market shares by Song and Xue [33] and Dong et al. [13] for the multinomial logit (MNL) model and by Li and Huh [25] for the nested logit (NL) model. Li and Huh [25] assumed that the price sensitivity parameters are the same across all products within the same nest. More recently, Gallego and Wang [16] relaxed this assumption and obtained a characterization of the optimal prices.

The above body of work either ignores price effects, focusing only on assortment effects, or adopts parametric models of price. Our work differs from the above literature in considering a nonparametric approach to model the joint effect of assortment and price. Further, most of the existing work focuses on optimization issues, with less emphasis on estimation. Our work addresses both estimation (from readily available transaction data) and optimization, resulting in an end-to-end solution.

Nonparametric approaches to choice modeling have been gaining traction within operations due to the increased availability of data. The prevailing nonparametric model describes customer purchase behavior using a general PMF on ranked lists of all the alternatives, including the no-purchase option. In the context of this general model, Farias and Jagabathula [15] proposed tractable procedures to predict expected revenues as a function of offer sets using historical sales transaction data. In van Ryzin and Vulcano [38], the authors proposed a "market discovery" algorithm, which constructs a preference PMF from sales data by iteratively finding ranked lists that increase the data log-likelihood value. Jagabathula [24] considers the problem of finding the revenue-maximizing offer set under a general rank-based choice model. They propose a general local search algorithm and show that the algorithm converges to the optimal solution in several important special cases. Finally, Honhon et al. [22] proposed efficient algorithms for assortment optimization for interesting structures of the general rank-based model. The above nonparametric approaches focus only on assortment effects of demand and do not model price effects. We extend the nonparametric rank-based demand model to also capture price effects and then solve the joint assortment and price optimization problem.

In marketing, our model has material connections to the literature on consideration sets. The idea that consumers use a two-stage decision process in which they first evaluate products to form a smaller relevant or consideration set and then make the purchase decision from this consideration set has been studied and empirically established in extensive work; see, for example, Hauser and Wernerfelt [21], Belonax and Mittelstaedt [6], Parkinson and Reilly [28], Alba and Chattopadhyay [2], Howard and Sheth [23], and Urban [36]. In particular, there are strong modeling connections in our work to Gilbride and Allenby [17], who demonstrated that a two-stage model in which customers pick a consideration set and then choose from the consideration set according to a discrete choice model has better in-sample and out-sample fit to conjoint study data on cameras, when compared to a discrete choice model alone.

2. Problem Formulation

In this section, we provide a precise description of our model and the problem formulation. We model the aggregate demand that a firm receives in response to the assortments and prices it offers. We consider a universe of n products, denoted by the set $\mathcal{N} = \{a_1, a_2, \ldots, a_n\}$, and suppose that the assortment is drawn from the product universe. Let a_0 denote the no-purchase or the outside option. Customers arrive sequentially to the firm and decide to either purchase one of the offered products or leave without making a purchase, in which case we say that the customer chooses the no-purchase option a_0. We allow for stockouts and price changes so that the offer set and prices seen by different customers may be different.

We suppose that customers use a two-stage model for making their purchase decision. In the first stage, a customer forms a *consideration set* of products by selecting the subset of offered products whose prices are less than or equal to a price threshold b. In the second stage, the customer chooses from the consideration set the most preferred product according to a preference list of the products. This model accounts for the phenomenon that customers use decision heuristics to simplify complicated decision tasks. Much of the existing work in marketing has provided empirical evidence for such a two-stage model; see, for example, Gilbride and Allenby [17] and the references therein.

Several screening rules have been studied in the existing literature. Commonly used screening rules are threshold based, resulting in the inclusion of products whose attribute values pass pre-specified thresholds. Because the only attribute that changes in our setting is price, we focus on the price-based screening rule. Without loss of generality, we suppose that the latent thresholds b belong to a finite set $\mathcal{B}$.

The population-level model is described by probability mass functions (PMFs) over price thresholds and preferences. Suppose products are offered at a price vector $\boldsymbol{p} = (p_a : a \in \mathcal{N})$, with p_a denoting the price of product a and $p_{a_0} = 0$. Then, the population is described by a threshold PMF $g\colon \mathcal{B} \to [0,1]$ and a preference PMF $\lambda_{\boldsymbol{p}}\colon \mathscr{S}_{n+1} \to [0,1]$, where $\mathscr{S}_{n+1}$ denotes the collection of all preference lists of the $n+1$ products $\mathcal{N} \cup \{a_0\}$. Each preference list $\boldsymbol{\sigma} = (\sigma_0, \sigma_1, \ldots, \sigma_n) \in \mathscr{S}_{n+1}$ specifies a rank ordering of the products, where σ_i is the product ranked at position i. For any product a, let $\sigma^{-1}(a)$ denote its preference rank. We suppose that lower ranked products are preferred to higher ranked products. When offered a subset $S \subseteq \mathcal{N}$ of products, the customer samples a threshold $b \in \mathcal{B}$ according to the threshold PMF g and a preference list $\boldsymbol{\sigma}$ according to the preference PMF $\lambda_{\boldsymbol{p}}$ and chooses to purchase the most preferred product from the set of products whose prices are less than or equal to b; that is, the customer's selection is given by $\arg\min_a \{\sigma^{-1}(a)\colon a \in S \cup \{a_0\}, p_a \le b\}$. Then, the choice probability $\theta_a(S, \boldsymbol{p})$ of choosing product $a \in S \cup \{a_0\}$ from the offer set S at price vector $\boldsymbol{p}$ is

$$\theta_a(S, \boldsymbol{p}) = \sum_{b \in \mathcal{B}} g_b \mathbb{P}_{\lambda_{\boldsymbol{p}}}(a \mid \{q \in S \cup \{a_0\}\colon p_q \le b\}),$$

where g_b denotes the probability of sampling threshold b under threshold PMF g, and $\mathbb{P}_{\lambda_{\boldsymbol{p}}}(a|C)$ denotes the probability that a preference list sampled according to $\lambda_{\boldsymbol{p}}$ results in the purchase of a from subset C:

$$\mathbb{P}_{\lambda_{\boldsymbol{p}}}(a \mid C) = \sum_{\boldsymbol{\sigma}} \lambda_{\boldsymbol{p}}(\boldsymbol{\sigma})\, \mathbb{1}[\boldsymbol{\sigma}, a, C], \quad \text{where} \quad \mathbb{1}[\boldsymbol{\sigma}, a, C] = \mathbb{1}_{\left\{\sigma^{-1}(a) \,<\, \sigma^{-1}(q) \ \forall\ q \in C \cup \{a_0\},\ q \neq a\right\}},$$

with $\mathbb{1}_{\{A\}}$ denoting the indicator variable that takes the value 1 whenever the event A is true and 0 otherwise. Note that $\mathbb{1}[\boldsymbol{\sigma}, a, C]$ is 1 if and only if product a has the smallest rank among all products in $C \cup \{a_0\}$.

The above two-stage model is very general and requires further definitions to be estimable from data. In particular, allowing the mapping $\boldsymbol{p} \mapsto \lambda_{\boldsymbol{p}}$ to be arbitrary results in a model that is not tractable and cannot extrapolate demand to new prices. To impose additional structure, we note that the model has two levers to capture the dependence of choice on prices: (a) the threshold b, which restricts the inclusion of a product in the consideration set, and (b) the preference PMF $\lambda_{\boldsymbol{p}}$, which captures any residual impact that price has on choice behavior. We suppose that this residual impact changes slowly with price. More precisely, we assume that the domain of the price vectors $\mathbb{R}_+^n$ is partitioned into K different regions, with $\mathbb{R}_+^n = \mathcal{P}_1 \cup \mathcal{P}_2 \cup \cdots \cup \mathcal{P}_K$ and $\mathcal{P}_k \cap \mathcal{P}_{k'} = \varnothing$ for $k \neq k'$. Each partition $\mathcal{P}_k$ is characterized by a preference PMF λ_k such that $\theta_a(S, \boldsymbol{p}) = \sum_{b \in \mathcal{B}} g_b \mathbb{P}_{\lambda_k}(a | \{q \in S \cup \{a_0\} : p_q \leq b\})$ for all $\boldsymbol{p} \in \mathcal{P}_k$. In other words, two price vectors that are "close enough" are assumed to result in the same preference PMF, suggesting that residual price impacts are slow to change with price. The complexity of the model increases linearly with the number of partitions K. As the amount of data increases, the number of partitions K may be increased, enriching the complexity of our model. Thus, the above partitioning eliminates the need for parametric assumptions on how $\lambda_{\boldsymbol{p}}$ varies with $\boldsymbol{p}$, resulting in a nonparametric dependence of $\lambda_{\boldsymbol{p}}$ on $\boldsymbol{p}$.

There is also empirical evidence to suggest that K is small. Specifically, our model of latent consideration sets is closely related to the work of Gilbride and Allenby [17], who considered a two-stage model of latent consideration sets and a multinomial probit model for preference lists. They fit this model to data collected from a conjoint study on customer preferences for a new camera format. They showed that the two-stage model has better in-sample and out-sample fit when compared to a discrete choice model alone. They also found strong evidence for the threshold-based screening rule. They [17, p. 11] stated that "Our results indicate that once the choice set[3] is formed, the price and body style do not play a role in the final decision." Based on this evidence, it is reasonable to assume that consideration set formation captures the primary effect of prices on choice. Any residual impact of price on choice is small, so that allowing the preference PMF to change slowly with prices should still result in good quality approximations while keeping the number of partitions K relatively small. Our numerical study provides empirical evidence supporting the quality of our approximations.

2.1. Discussion of model assumptions

Relation to existing approaches in operations. Our two-stage model class extends the rank-based choice model proposed in Farias and Jagabathula [15]. The rank-based choice model in uses only rank-orderings of products as model primitives. It is very general, subsumes the random utility maximization (RUM) class of models, and removes the need for potentially unrealistic, structural assumptions about how utilities are generated. However, despite its flexibility in capturing a wide-range of choice behaviors, it cannot predict choices when prices or other product features vary. Such predictions require explicit modeling of how the rank orderings vary in response to variation in product prices. Utility-based models – such as the popular MNL, NL, and latent-class MNL

[3] The authors use "choice set" for "consideration set."

(LC-MNL) models – capture the dependence of choices on prices by modeling product utilities as functions of product prices. Such an approach results in models that are either parsimonious and tractable (such as the MNL and NL models) or flexible (such as the LC-MNL model) but *not both*. Our approach is to retain flexibility by operating in the space of rank orderings of products and to introduce structure via the layer of consideration sets. This approach allows us to retain the appealing properties of rank-based choice models as well as leverage existing methods to estimate such models.

Relation to the literature on consideration sets. The idea that consumers use a two-stage decision process in which they first evaluate products to form a smaller consideration set and then make the purchase decision from this consideration set has been studied and empirically verified through extensive work in marketing (see the references cited at the end of Section 1.1). Our price-based screening rule belongs to the class of conjunctive decision rules, which have been studied and empirically established in the marketing literature (see Hauser and Dzyabura [20]). A conjunctive rule specifies that a customer considers a product only if the attribute values of *all* the attributes are above (or below) certain acceptable thresholds (and thus conjunctive). The latent price threshold in our model corresponds to the acceptable level for price. As we only allow product prices to vary (with all the other attributes remaining fixed), products that may not be acceptable due to other attributes can equivalently be assumed to be less preferred than the no-purchase option. Therefore, our screening rule captures general conjunctive screening rules.

Operational tractability. Finally, our modelling assumptions are motivated by the desire to strike a balance between the flexibility of the models and their tractability. However, it is not immediately clear if the high dimensional distribution λ over preference lists (whose dimension scales with $n! \sim (n/e)^n$ where n is the number of products) lends itself to tractable estimation and optimization. Surprisingly, we identify a structure in the data that allows us to obtain a continuous trade-off between the "complexity" of the data and the tractability of the estimation and optimization problems. To the best of our knowledge, our work is the first to provide theoretical guarantees for rank-based choice models that operate directly in the space of distributions over preference lists.

2.2. Data model

We assume data that are available to us in the form of a sequence of choices made by the customers in response to particular offer sets and prices. Formally, we assume that we are given choices of T customers in the form of tuples $\mathsf{Data} = \{(c_1, S_1, \boldsymbol{p}_1), (c_2, S_2, \boldsymbol{p}_2), \ldots, (c_T, S_T, \boldsymbol{p}_T)\}$, where $c_t \in S_t \cup \{a_0\}$ is the product chosen by customer t when offered products in subset S_t at prices $\boldsymbol{p}_t =$

$(p_{at} : a \in \mathcal{N})$. We assume that $p_{at} = +\infty$ for any $a \notin S_t$. As the price vectors also contain information about the products that are not offered (because we set their prices to infinity), we sometimes simply write $\mathsf{Data} = \{(c_1, \boldsymbol{p}_1), (c_2, \boldsymbol{p}_2), \ldots, (c_T, \boldsymbol{p}_T)\}$ for brevity of notation.

The type of data that we assume is typically readily available in practice either in the form of purchase transactions or aggregated market shares. Because we are assuming that each transaction corresponds to a different customer, aggregated market share data can be readily transformed into the form above by creating a dummy customer for each product purchase (where the number of product purchases is determined by multiplying the market share by the market size). In the data specification above, we assume that we also observe the selection of the no-purchase option or, equivalently, the size of the market. This is a standard assumption, and one can adopt any of the several demand untruncation methods proposed in the literature (see Haensel and Koole [18] and the references therein) to deal with the data censoring issue. We discuss one such adaptation in Appendix F.

2.3. Overview of research questions

The biggest challenge with our modeling framework is its computational tractability. To address this challenge, we ask the following two questions: 1) How can we tractably fit the model to transaction data and then determine the optimal joint assortment and pricing decision? 2) How does the computational complexity depend on the complexity of the pricing structure? Answering the first question makes our modeling framework *operational* by providing algorithms to estimate parameters and then determine the optimal decision. Answering the second question allows a manager to make a continuous trade-off between the computational burden and the flexibility afforded by complex pricing structures.

We answer the above questions in three steps; we (a) develop a general purpose estimation methodology to fit our model to historical transactions, (b) identify tractable pricing structures and relate the complexity of the pricing structure to the computational burden of estimation, and (c) extend the tractability of the identified pricing structures to joint assortment and price optimization.

Estimation methodology: We use the method of maximum likelihood estimation (MLE) to estimate our model parameters. Our model is described by a partition of the domain of the price vectors and the parameters λ_k and g_k for each partition. Joint estimation of the partitions and model parameters is challenging. In fact, even when the partitions are given, estimating model parameters is an NP-complete problem (cf. Proposition 4.1). As a result, we focus the estimation section on estimating model parameters when partitioning is given. We discuss techniques to embed our estimation methodology into a partitioning scheme in the numerical experiments in Section 6. With a given partitioning, model parameters can be estimated separately for each partition. We focus on an arbitrary partition and drop the partition subscripts from λ and g to simplify notation.

Assuming that each data point is generated from an independent draw from the model, the log likelihood of the data is given by

$$\mathcal{L}\left(g, \lambda \mid \mathsf{Data}\right) = \sum_{t=1}^{T} \log\left(\sum_{b \in \mathcal{B}} g_b \mathbb{P}_\lambda(c_t | \{a \in S_t \cup \{a_0\} : p_a \le b\})\right).$$

We select as estimates, the parameters $\hat{g}$ and $\hat{\lambda}$ that maximize the log-likelihood function:

$$\hat{g}, \hat{\lambda} \in \arg\max_{g,\lambda} \mathcal{L}\left(g, \lambda \mid \mathsf{Data}\right) = \arg\max_{g,\lambda} \sum_{t=1}^{T} \log\left(\sum_{b \in \mathcal{B}} g_b \mathbb{P}_\lambda(c_t | \{a \in S_t \{a_0\} : p_a \le b\})\right). \tag{1}$$

We face several challenges in solving the MLE optimization problem in (1). We discuss these issues in detail in Section 3.

Tractable pricing structures. The computational complexity of our estimation procedure is dominated by the complexity of solving a high-dimensional linear program (LP) with $O(n!)$ variables. We show that the LP reduces to the rank aggregation problem of determining a ranking $\boldsymbol{\sigma}$ that minimizes a linear cost function, which is NP-complete (cf. Proposition 4.1). However, what is not clear is what the source of complexity is. Intuitively, we expect the complexity of estimation to depend on the complexity of the data. But how does one characterize the complexity of the data? We answer this question by identifying classes $\mathscr{P}_d$ of pricing structures, termed the d-sorted pricing structures, where d is an integer taking values between 0 and n. These pricing structures are nested: $\mathscr{P}_0 \subset \mathscr{P}_1 \subset \mathscr{P}_2 \subset \cdots \subset \mathscr{P}_{n-1} \subset \mathscr{P}_n = \mathbb{R}^n_+$, so that d captures the complexity of the pricing structure. Assuming that all the training prices belong to $\mathscr{P}_d$, we show that the complexity of solving the LP is polynomial in n for a fixed d. We discuss the details in Section 4.

Operational tractability: After estimating the parameters of the model, we consider the canonical operational problem of determining the combination of offer set and price vector that maximizes the expected revenue, corresponding to the following optimization problem:

$$\max_{S \subseteq \mathcal{N},\ \boldsymbol{p} \in \mathbb{R}^n_+} R(S, \boldsymbol{p}) = \max_{S \subseteq \mathcal{N},\ \boldsymbol{p} \in \mathbb{R}^n_+} \sum_{a \in S} p_a \theta_a(S, \boldsymbol{p}). \tag{2}$$

The decision problem in (2) is computationally challenging to solve because it not only requires searching over all possible subsets, but the revenue function is also globally non-concave in prices even for a fixed subset S. The above problem, in general, is NP-hard. Surprisingly, we show that if we restrict the prices to the d-sorted family $\mathscr{P}_d$ for a fixed d, then there exists a polynomial-time approximation scheme for finding the optimal solution. We discuss the details of the decision problem in Section 5.

3. Estimation Methodology

As mentioned, we estimate the model parameters by solving the MLE problem in (1). This problem is in general non-concave and involves a factorial number of variables, making it computationally challenging to solve to optimality. As a result, we settle for finding a local maximum or a stationary point.

To find a stationary point, we use the EM meta-heuristic of Dempster et al. [11]. The EM heuristic is commonly used to simplify the maximization of log-likelihood functions when some of the variables are latent, but the log-likelihood function becomes globally concave when the latent variables are observed. In our setting, the price thresholds are latent, but the log-likelihood function becomes globally concave when the price thresholds are observed; see Lemma 3.1 below, proved in Appendix A.1.

LEMMA 3.1. *Let $\mathcal{C} = \{(a, A) \colon a = c_t$ and $A = S_t^b$ for some $b \in \mathcal{B}$ and $1 \le t \le T\}$. The complete data log-likelihood function is concave and separable in the parameters g and λ, and is given by*

$$\mathcal{L}_C = \sum_{b \in \mathcal{B}} m_b \log g_b + \sum_{(a,A) \in \mathcal{C}} \gamma_{a,A} \log \mathbb{P}_\lambda(a|A) \ , \tag{3}$$

where m_b denotes the number of customers in the data with threshold b and $\gamma_{a,A}$ denotes the number of customers t with the choice and consideration set pair (a, A).

However, because the price thresholds, and hence $(m_b : b \in \mathcal{B})$ and $(\gamma_{a,A} : (a, A) \in \mathcal{C})$, are not observed, the complete data log-likelihood function in (3) cannot be directly optimized. The EM method deals with this issue as follows. It starts with arbitrary initial estimates of the model parameters $\hat{g}$ and $\hat{\lambda}$ and computes the conditional expected values $\mathbb{E}[\mathcal{L}_C|\hat{g}, \hat{\lambda}]$ (the E-step). It then maximizes the resulting conditional likelihood function to generate new estimates of the model parameters (the M-step). The two steps are carried out iteratively with the model parameter estimates generated in each step used as inputs for the next step. The iterations are carried out until convergence. Lemma 3.2 shows that the E-step can be performed efficiently.

LEMMA 3.2 **(Conditional expectation for the E-step)**. *Given model parameter estimates $\hat{g}$ and $\hat{\lambda}$, the conditional expectation of the complete data log-likelihood function is*

$$\mathbb{E}[\mathcal{L}|\hat{g}, \hat{\lambda}] = \sum_{b \in \mathcal{B}} \hat{m}_b \log g_b + \sum_{(a,A) \in \mathcal{C}} \hat{\gamma}_{a,A} \log \mathbb{P}_\lambda(a|A), \tag{4}$$

where $\hat{m}_b$ is the expected number of customers with latent threshold b and $\hat{\gamma}_{a,A}$ is the expected number of customers with the choice and consideration set pair (a, A), given by

$$\hat{m}_b = \sum_{t=1}^{T} h_t(b) \quad \text{and} \quad \hat{\gamma}_{a,A} = \sum_{t=1}^{T} \sum_{b \in \mathcal{B}} h_t(b) \mathbb{1}_{\{a = c_t, A = S_t^b\}} \ , \tag{5}$$

where $h_t(b)$ is the probability that customer t has latent threshold b conditioned on her observation $(c_t, S_t, \boldsymbol{p}_t)$ and is given by $h_t(b) = \hat{g}_b \mathbb{P}_{\hat{\lambda}}(c_t|S_t^b) / \sum_{b' \in \mathcal{B}} \hat{g}_{b'} \mathbb{P}_{\hat{\lambda}}(c_t|S_t^{b'})$.

Based on the result in Lemma 3.2, the EM procedure may be summarized as follows.

Overview of the EM procedure

Step 0. [Initialization] Set $\hat{g}$ and $\hat{\lambda}$ to arbitrary feasible initial values.

Step 1. [E-step] Compute the following estimates using (5):

$$\hat{m}_b = \text{ expected \# of customers with latent threshold } b \text{ for } b \in \mathcal{B}, \text{ and}$$

$$\hat{\gamma}_{a,A} = \text{ expected \# of customers with choice, a consideration set tuple } (a, A) \text{ for } (a, A) \in \mathcal{C}.$$

Step 2. [M-step] Generate new estimates $\boldsymbol{g}^*$ and λ^* as follow:

$$\boldsymbol{g}^* = \underset{\boldsymbol{g} \geq \mathbf{0}: \sum_{b \in \mathcal{B}} g_b = 1}{\arg\max} \sum_{b \in \mathcal{B}} \hat{m}_b \log g_b = \left(\frac{\hat{m}_b}{\sum_{q \in \mathcal{B}} \hat{m}_q} : b \in \mathcal{B} \right)$$

$$\lambda^* = \arg\max \left\{ \sum_{(a,A) \in \mathcal{C}} \hat{\gamma}_{a,A} \log \mathbb{P}_\lambda(a|A) \ : \ \sum_{\boldsymbol{\sigma} \in \mathscr{S}_{n+1}} \lambda(\boldsymbol{\sigma}) = 1, \ \lambda(\boldsymbol{\sigma}) \geq 0 \ \ \forall \boldsymbol{\sigma} \right\}.$$

Step 3. [Check stopping condition] If the stopping condition is not met, then $\hat{g} \leftarrow g^*$ and $\hat{\lambda} \leftarrow \lambda^*$ and go to *Step 1*. Otherwise, terminate with the output g^* and λ^*.

Several stopping conditions are appropriate; we discuss the issue in Section 3.3.

The E-step in the above algorithm can be carried out efficiently because it only relies on $O(nT)$ choice probabilities, $\mathbb{P}_{\hat{\lambda}}(a|A)$ for all $(a, A) \in \mathcal{C}$, as opposed to the entire $(n+1)!$ dimensional distribution $\hat{\lambda} : \mathscr{S}_{n+1} \to [0, 1]$. However, carrying out the M-step requires solving the following optimization problem:

$$\max \left\{ \sum_{(a,A) \in \mathcal{C}} \hat{\gamma}_{a,A} \log \mathbb{P}_\lambda(a|A) \ : \ \sum_{\boldsymbol{\sigma} \in \mathscr{S}_{n+1}} \lambda(\boldsymbol{\sigma}) = 1, \ \lambda(\boldsymbol{\sigma}) \geq 0 \ \ \forall \boldsymbol{\sigma} \right\}. \tag{M-step}$$

This problem presents two key challenges: (a) multiplicity of optimal solutions and (b) factorial number of variables. We discuss how we address each of these challenges next.

3.1. Multiplicity of optimal solutions: Reparametrization of the M-step

The optimization problem M-step has multiple optimal solutions because the available data are not sufficient to allow for point identification of the model parameters. This multiplicity is a result of the flexibility we have afforded our model and stems from the fundamental non-identifiability of distributions over rankings from choice data. To see this, let λ^* be an optimal solution and consider choice probabilities $y^*_{a,A} \stackrel{\text{def}}{=} \mathbb{P}_{\lambda^*}(a|A)$. Sher et al. [32] show that for $n \geq 4$, there are multiple PMFs over preference lists that are consistent with *any* given collection of choice probabilities. A rough proof follows from the observation that while the choice probabilities impose $O(2^n)$ degrees of freedom, the underlying distribution over preference lists has $O(n!) = O(2^{n \log n})$ degrees of freedom.

We overcome the multiplicity issue through a reparametrization of the $\mathsf{M\text{-}step}$ that provides a compact description of the set of optimal solutions. Because we only need the inputs $\hat{g}$ and the choice probabilities $\mathbb{P}_{\hat{\lambda}}(a|A)$ for all $(a,A) \in \mathcal{C}$ to carry out each EM iteration, we consider the following reformulation:

$$\max\left\{\sum_{(a,A)\in\mathcal{C}} \hat{\gamma}_{a,A} \log y_{a,A} \;:\; y_{a,A} = \mathbb{P}_{\lambda}(a|A) \quad \forall\,(a,A)\in\mathcal{C}, \sum_{\boldsymbol{\sigma}\in\mathscr{S}_{n+1}} \lambda(\boldsymbol{\sigma}) = 1,\ \lambda(\boldsymbol{\sigma}) \geq 0 \ \ \forall \boldsymbol{\sigma}\right\}. \qquad (6)$$

The optimal solution $\boldsymbol{y}^*$ encapsulates *all* the optimal solutions λ^* to the $\mathsf{M\text{-}step}$, $\left\{\lambda \colon y^*_{a,A} = \mathbb{P}_{\lambda}(a|A) \,\forall\,(a,A)\in\mathcal{C}\right\}$, and hence is sufficient to carry out the E-step in the next iteration. Depending on the subsequent prediction or decision problem, a specific distribution λ can be chosen from the identified set.

3.2. Factorial number of variables: Obtaining an improving solution for $\mathsf{M\text{-}step}$

With the reformulation above, the $\mathsf{M\text{-}step}$ optimization problem can be written more succinctly as

$$\max_{y\in Q_{\mathcal{C}}} f(\boldsymbol{y}) \stackrel{\text{def}}{=} \sum_{(a,A)\in\mathcal{C}} \hat{\gamma}_{a,A} \log y_{a,A}, \qquad (7)$$

where the polytope $Q_{\mathcal{C}}$ is defined as

$$Q_{\mathcal{C}} \stackrel{\text{def}}{=} \left\{\boldsymbol{y} \in [0,1]^{|\mathcal{C}|} \;:\; y_{a,A} = \mathbb{P}_{\lambda}(a|A) \ \ \forall\,(a,A)\in\mathcal{C}, \sum_{\boldsymbol{\sigma}\in\mathscr{S}_{n+1}} \lambda(\boldsymbol{\sigma}) = 1,\ \lambda(\boldsymbol{\sigma}) \geq 0 \ \ \forall \boldsymbol{\sigma}\right\}. \qquad (8)$$

The presence of a factorial number of variables makes it computationally challenging to solve (7), despite it being a concave maximization problem. We overcome this issue by accomplishing the simpler task of obtaining an improving solution $\boldsymbol{y}^*$, whose objective value is greater than or equal to the objective value at the existing estimate $\hat{\boldsymbol{y}}$. This relaxed variant of the EM algorithm, which relies only on finding an improving solution to the M-step, is called the generalized EM algorithm and can be shown to converge to a stationary point of the log-likelihood function.

To find an improving solution, we find a surrogate function that is easier to optimize. Because linear functions are generally tractable, we use a local linear approximation of the objective function in (7) as the surrogate. In order to simplify notation, let $f(\boldsymbol{y})$ denote the objective function $\sum_{(a,A)\in\mathcal{C}} \hat{\gamma}_{a,A} \log y_{a,A}$ of the optimization problem in (7). Then, we can establish the following result:

PROPOSITION 3.1 **(Certificate of optimality for the M-step).** *Let $\hat{\boldsymbol{y}} \in Q_{\mathcal{C}}$ be given, and let $\boldsymbol{x}^*$ denote an optimal solution to the linear program*

$$\max_{\boldsymbol{x} \in Q_{\mathcal{C}}} \sum_{(a,A) \in \mathcal{C}} c_{a,A} x_{a,A} \ , \qquad (\text{M-step LP})$$

where $c_{a,A} = \hat{\gamma}_{a,A}/\hat{y}_{a,A}$ for all $(a,A) \in \mathcal{C}$. If $\sum_{(a,A)\in\mathcal{C}} c_{a,A} x^*_{a,A} \le T$, then $\hat{\boldsymbol{y}}$ is the optimal solution to (7). On the other hand, if $\sum_{(a,A)\in\mathcal{C}} c_{a,A} x^*_{a,A} > T$, then there exists an $\alpha \in (0,1)$ such that for $\boldsymbol{y} = \alpha\hat{\boldsymbol{y}} + (1-\alpha)\boldsymbol{x}^*$, we have $f(\boldsymbol{y}) > f(\hat{\boldsymbol{y}})$. Such an improving solution $\boldsymbol{y}$ may be found using a one-dimensional line search

$$\max_{\alpha \in [0,1]} f(\alpha\hat{\boldsymbol{y}} + (1-\alpha)\boldsymbol{x}^*),$$

which can be done efficiently because the function $\alpha \mapsto f(\alpha\hat{\boldsymbol{y}} + (1-\alpha)\boldsymbol{x}^*)$ is strictly concave on [0,1].

Proposition 3.1 provides a certificate of optimality for the optimization problem in (7). The certificate requires solving the M-step LP. If the optimal value of the LP is less than or equal to the total number of customers T, then $\hat{\boldsymbol{y}}$ is the optimal solution to (7), and therefore we cannot find a solution that increases the objective value of (7) beyond that at $\hat{\boldsymbol{y}}$. On the other hand, if the optimal objective value turns out to be strictly bigger than T, then there always exists a solution $\boldsymbol{y}$ that strictly increases the objective value $f(\cdot)$ beyond the current solution $\hat{\boldsymbol{y}}$. Such an improving solution may be found using a one-dimensional line search $\max_{\alpha\in[0,1]} f(\alpha\hat{\boldsymbol{y}} + (1-\alpha)\boldsymbol{x}^*)$, which can be done efficiently because $\alpha \mapsto f(\alpha\hat{\boldsymbol{y}} + (1-\alpha)\boldsymbol{x}^*)$ is strictly concave on $[0,1]$; see, for example, Boyd and Vandenberghe [9]. We note that although the M-step LP in Proposition 3.1 involves $n!$ variables $\{\lambda(\boldsymbol{\sigma}) : \boldsymbol{\sigma} \in \mathscr{S}_{n+1}\}$, the feasible polytope $Q_{\mathcal{C}}$ has many interesting properties that can be exploited for efficient computation. We defer these issues to Section 4.

3.3. Putting everything together

We have the following convergence result for the above EM algorithm:

THEOREM 3.1 **(Convergence of parameter estimates).** *For all k,*

$$\mathcal{L}\left(g^{(k)}, \boldsymbol{y}^{(k)}\right) < \mathcal{L}\left(g^{(k+1)}, \boldsymbol{y}^{(k+1)}\right) \ ,$$

and thus, the sequence of the log-likelihoods associated with the parameter estimates generated by the EM algorithm converges to a value corresponding to a stationary point of the log-likelihood function.

The proof is standard and follows from standard EM machinery (Dempster et al. [11]), so we omit the details. Appendix A.4 presents a formal description of the EM algorithm. We make a few remarks about the implementation:

- *Stopping criteria.* We terminate the algorithm when the increase in the log-likelihood value is within a pre-specified tolerance parameter.
- *Initialization.* The initialization can be arbitrary as long as the resulting choice probabilities $y_{a,A}$ are strictly positive for all $(a,A)\in\mathcal{C}$. We describe a greedy initialization algorithm in the appendix (cf. Algorithm 1 in Appendix D) that initializes $\lambda^{(0)}$ to have a support of size $n+1$ and strictly positive choice probabilities for all $(a,A)\in\mathcal{C}$.
- *Uniqueness of the solution to* (6). The optimization problem in (6) has a unique solution if $\hat{\gamma}_{a,A}>0$ for all $(a,A)\in\mathcal{C}$ because $\log(\cdot)$ is strictly concave. It can be seen from Lemma 3.2 that $\hat{\gamma}_{a,A}>0$ for all $(a,A)\in\mathcal{C}$ if $\hat{g}_b>0$ for all $b\in\mathcal{B}$. We assume that the EM method is started with an initial estimate g that has a positive probability mass over all the thresholds in $\mathcal{B}$. This ensures that $\hat{g}_b>0$ for all $b\in\mathcal{B}$ in all the iterations, so that $\hat{\gamma}_{a,A}>0$ for all $(a,A)\in\mathcal{C}$.
- *Extension to data with unobserved choice of the no-purchase option:* The EM algorithm in this section assumes that we observe the choice of the no-purchase option in the dataset. However, in many settings, the sales transaction data do not record the choice of the no-purchase option. We extend our proposed EM method to handle the missing observations of the choice of the no-purchase option by using the approach of Vulcano and van Ryzin [37]. The details of the extension are given in Appendix F.

Additional implementation details are discussed in Section 6 on numerical studies.

4. An Efficient Solution of the M-step LP and d-sorted Pricing Structures

In this section, we focus on efficiently solving the M-step LP in Proposition 3.1. For that, we identify classes of pricing structures of increasing complexity and relate the complexity of the pricing structure to the computational complexity of solving the M-step LP

$$\max_{\boldsymbol{x}\in Q_{\mathcal{C}}} \sum_{(a,A)\in\mathcal{C}} c_{a,A}x_{a,A}\ ,$$

where the polytope $Q_{\mathcal{C}}$ is given in (8). The optimal solution of the above LP will occur at an extreme point. Because $Q_{\mathcal{C}}$, by definition, is the convex hull of the $(n+1)!$ points $\boldsymbol{e_\sigma}\in\{0,1\}^{|\mathcal{C}|}$ defined as $e_{\boldsymbol{\sigma},a,A}=\mathbb{1}[\boldsymbol{\sigma},a,A]$, every extreme point must be equal to $\boldsymbol{e_\sigma}$ for some $\boldsymbol{\sigma}\in\mathscr{S}_{n+1}$. As a result, solving the M-step LP is equivalent to solving the following optimization problem:

$$\max_{\boldsymbol{\sigma}\in\mathscr{S}_{n+1}} \sum_{(a,A)\in\mathcal{C}} c_{a,A}\mathbb{1}[\boldsymbol{\sigma},a,A]. \tag{9}$$

The optimization problem above has the following intuitive interpretation: for each choice and consideration set pair (a,A), the weight $c_{a,A}$ quantifies the importance of that pair. Our goal is to find the ranking that is "most consistent" with the pairs (a,A) according to weights $c_{a,A}$.

The optimization problem in (9) is similar to the linear program obtained in recent work by van Ryzin and Vulcano [38] when fitting distributions over rankings using the maximum likelihood method. It is also similar to the problems that have appeared in the existing machine learning literature (see Ali and Meilă [3] and Meilă et al. [27]) in the context of *rank aggregation*, where the goal is to find the ranking that is most consistent with a given set of rankings, under appropriate definitions of consistency. Particularly, as shown in Proposition 4.1, a special case of (9) is the popular *Kemeny optimization problem*, in which the objective is to find a single ranking that minimizes the average distance from a given collection of total orderings/rankings over n items.

PROPOSITION 4.1 **(Hardness of solving the M-step LP).** *The* M-step LP *is equivalent to* $\max_{\boldsymbol{\sigma} \in \mathscr{S}_{n+1}} \sum_{(a,A) \in \mathcal{C}} c_{a,A} \mathbb{1}[\boldsymbol{\sigma}, a, A]$, *and it is NP-complete. Kemeny optimization, however, is NP-hard [3], which implies that* (9) *is also NP-hard.*

Proposition 4.1 is proved in Appendix B. While Kemeny optimization has been well studied in the literature, the M-step LP itself has received little attention. To the best of our knowledge, there has been no study of the source of complexity in solving the M-step LP. Our goal is to understand the *source of complexity* and understand whether there are special and practically relevant cases that can be solved efficiently. We discuss this next.

4.1. The d-sorted pricing structure

To characterize the complexity of solving the M-step LP, we introduce the class of d-sorted pricing structures. To formally define these pricing structures, consider a specific ordering $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_n)$ of the products $\{a_1, a_2, \ldots, a_n\}$, and for all i, let $\tau^{-1}(a_i) \in \{1, 2, \ldots, n\}$ denote the rank of product a_i under $\boldsymbol{\tau}$. Define the collection of price vectors $\mathscr{P}_d$ as follows:

$$\mathscr{P}_d = \left\{ \boldsymbol{p} \in \mathbb{R}_+^n : \; \left| \pi_{\boldsymbol{p}}^{-1}(a_i) - \tau^{-1}(a_i) \right| \le d \;\forall\; i \text{ such that } p_i \ne +\infty \right\}, \tag{10}$$

where d is an integer such that $0 \le d \le n$ and $\pi_{\boldsymbol{p}}$ represent the price ordering of the products according to price vector $\boldsymbol{p}$ so that $p_{\pi_{\boldsymbol{p}}(1)} \le p_{\pi_{\boldsymbol{p}}(2)} \le \cdots \le p_{\pi_{\boldsymbol{p}}(n)}$ with $\pi_{\boldsymbol{p}}^{-1}(a_i)$ denoting the rank of product indexed i and $\pi_{\boldsymbol{p}}(i)$ denoting the product that is ranked at rank i under $\boldsymbol{p}$. Also, recall that we assume $\boldsymbol{p}_i = \infty$ if product i is not offered. By definition,

$$\mathscr{P}_0 \subset \mathscr{P}_1 \subset \mathscr{P}_2 \subset \cdots \subset \mathscr{P}_{n-1} \subset \mathscr{P}_n = \mathbb{R}_+^n \;.$$

When $d = 0$, $\mathscr{P}_0$ is the set of price vectors that have exactly the same ordering; that is,

$$\mathscr{P}_0 = \left\{ \boldsymbol{p} \in \mathbb{R}_+^n : p_{\tau_1} \le p_{\tau_2} \le \cdots \le p_{\tau_n} \right\} \;.$$

This type of constraint ($d=0$) on the price vectors is quite popular and has been been shown to result in tractable estimation and optimization problems (cf. Rusmevichientong [30] and Aggarwal et al. [1]). For $d \geq 1$, the set $\mathscr{P}_d$ can be thought of as a generalization of a strict sorting constraint with the price ranks of the products allowed to deviate from their corresponding reference ranks by no more than d. When $d=n$, it is clear that $\mathscr{P}_d$ admits all possible price vectors so that $\mathscr{P}_n = \mathbb{R}_+^n$. Thus, $\mathscr{P}_d$ defines collections of price vectors of increasing complexity.

In practice, d-sorted pricing structures with small values of d arise when the firm selects a base price ordering $\boldsymbol{\tau}$ and offers prices that are generally consistent with the base ordering but are allowed to deviate by an amount d. Such sorted price structures are generally reasonable when the products are vertically differentiated by brand, price, quality, etc. They also arise from a firm's business constraints, such as premium brands always being priced above non-premium ones. To our knowledge, these pricing structures have received little attention in the literature. But, as we show below, the class of price vectors $\mathscr{P}_d$ possesses structure that we can exploit to solve both the estimation and optimization problem efficiently with the computational complexity that is polynomial in n for a fixed d but exponential in d. When d is small, our algorithms are guaranteed to be efficient. For the real-world dataset used in our numerical experiments in Section 6, we observed that the product prices in the transaction data possess the d-sorted pricing structure with values of d around 3 or 4, depending on the product category.

To state our result for estimation, suppose that all the offered price vectors belong to $\mathscr{P}_d$ for a particular reference ordering $\boldsymbol{\tau}$. Define the collection of tuples

$$\mathcal{C}_d = \{(a, A) \colon a \in A, A = \{a' \colon p_{a'} \leq b\} \text{ for some } b \in \mathcal{B}, \boldsymbol{p} \in \mathscr{P}_d\}, \tag{11}$$

comprising of all possible choice and consideration set combinations when the offered price vectors belong to $\mathscr{P}_d$. It follows from the definition of $\mathscr{P}_d$ that $\mathcal{C}_0 \subseteq \mathcal{C}_1 \subseteq \cdots \subseteq \mathcal{C}_n$ and $\mathcal{C}_n$ is the set of all possible subsets of the n products. With this definition, we can establish that the rank aggregation problem (9), and consequently the $\mathsf{M\text{-}step\ LP}$, can be solved efficiently in n for a fixed d. Specifically, we have the following result:

Proposition 4.2 **(Efficient solution of the $\mathsf{M\text{-}step\ LP}$ through DP).** *Consider the following rank aggregation problem:*

$$\max_{\boldsymbol{\sigma} \in \mathscr{S}_{n+1}} \sum_{(a,A) \in \mathcal{C}} c_{a,A} \mathbb{1}[\boldsymbol{\sigma}, a, A] .$$

If $\mathcal{C} \subseteq \mathcal{C}_d$, then the problem can be solved via a DP in $O(n^3 4^d |\mathcal{C}|)$ operations.

Note that, by definition, $|\mathcal{C}| = O(nT)$, and thus, for small d, the $\mathsf{M\text{-}step\ LP}$ can be solved efficiently. The detailed proof of Proposition 4.2 is given in Appendix B.2. We provide a brief proof sketch here for the special case where $d=0$. The same argument can be extended for general d.

Proof sketch of Proposition 4.2 for $d=0$: Without loss of generality, assume that $\boldsymbol{\tau}$ is the identity ordering, so $\mathscr{P}_0=\{\boldsymbol{p}: p_1\leq\cdots\leq p_n\}$. With the constraint that all the training price vectors are in $\mathscr{P}_0$, it can be seen that $\mathcal{C}=\{(i,\{a_0,\ldots,a_j\})\colon 0\leq i\leq j\leq n\}$. To simplify notation, let c_{ij} denote $c_{a_i,A}$ for $A=\{a_0,\ldots,a_j\}$. Also, for any ranking $\boldsymbol{\sigma}$, let σ_{ij} denote the indicator variable taking the value of 1 if and only if $\sigma^{-1}(a_i)<\sigma^{-1}(a_k)$ for all $0\leq k\leq j$, $k\neq i$. Finally, let $c_{00}=0$. With this notation, (9) becomes

$$\max_{\boldsymbol{\sigma}\in\mathscr{S}_{n+1}}\sum_{j=0}^{n}\sum_{i=0}^{j}c_{ij}\sigma_{ij}.$$

We now propose a DP to recursively construct the optimal preference list. For any ranking $\boldsymbol{\sigma}=(\sigma_0,\sigma_1,\ldots,\sigma_n)$, let σ_r denote the product ranked at position r. The objective in the above optimization problem can be shown to be $\sum_{r=0}^{n}\sum_{j=\sigma_r}^{n}c_{ij}\sigma_{\sigma_r j}$. Now, because σ_r is at position r, $\sigma_{\sigma_r j}$ is 1 if and only if none of the products ranked below r are part of the set $\{a_0,\ldots,a_j\}$. We thus have

$$\sum_{j=0}^{n}\sum_{i=0}^{j}c_{ij}\sigma_{ij}=\sum_{r=0}^{n}\sum_{j=\sigma_r}^{\min\{\sigma_0,\sigma_1,\ldots,\sigma_{r-1}\}-1}c_{\sigma_r j}.$$

The above expression of the objective function suggests a DP in which we construct the optimal ranking sequentially from the least preferred product to the most preferred product. In order to facilitate the formulation of the DP, we represent each ranking $\boldsymbol{\sigma}$ as the collection of tuples $(\sigma_0,\xi_0),(\sigma_1,\xi_1),\ldots,(\sigma_n,\xi_n)$, where, as before, σ_r denotes the product ranked at position r by $\boldsymbol{\sigma}$ and $\xi_r=\min\{\sigma_0,\sigma_1,\ldots,\sigma_{r-1}\}$ for any $r\geq 1$. For completeness, we set $\xi_0=n+1$. With the above representation of $\boldsymbol{\sigma}$, the objective function can be written as $\sum_{r=0}^{n}\sum_{j=\sigma_r}^{\xi_r-1}c_{\sigma_r j}$.

It is clear from the above expression that the choice of σ_r is *only* influenced by ξ_r, which is determined by the products ranked at the previous $r-1$ positions and will influence the choice of products to be ranked at positions beyond r *only* through ξ_{r+1}. As a result, we define a DP value function at stage r with the state variable ξ_r. Specifically, we define for any $0<r\leq n$ and $0\leq\xi\leq n-r+1$ the value functions

$$V_r(\xi)=\begin{cases}\max_{\sigma_r}\left[\sum_{j=\sigma_r}^{\xi-1}c_{\sigma_r j}+V_{r+1}(\min\{\sigma_r,\xi\})\right] & \text{if } \xi<n-r+1,\\ \max_{0\leq\sigma_r\leq\xi-1}\left[\sum_{j=\sigma_r}^{\xi-1}c_{\sigma_r j}+V_{r+1}(\sigma_r)\right] & \text{if } \xi=n-r+1\end{cases}$$

with the boundary condition $V_{n+1}(\xi)=0$ for all $1\leq\xi\leq n$ and $V_0(n+1)=\max_{0\leq\sigma_0\leq n}\left[\sum_{j=\sigma_0}^{n}c_{\sigma_0 j}+V_1(\sigma_0)\right]$. Note that we need to compute a total of $O(n)$ value functions, each of dimension $O(n)$. Computing each value function requires us to search over $O(n)$ functions to find the maximum value, and computing each of those values requires $O(n)$ computations. Therefore, the running time of the above DP is $O(n^4)$. It can be shown (cf. Proposition 4.2) that solving the DP results in a feasible solution. The above argument can be extended to the case when all the price vectors belong to the set $\mathscr{P}_d$. Details are provided in Appendix B.2.

5. Operational Tractability: Joint Assortment and Price Optimization

We now focus on making the decision: determining the offer-set and price combination to maximize revenues (that is, solving $\max_{S \subseteq \mathcal{N}, \boldsymbol{p} \in \mathbb{R}_+^n} R(S, \boldsymbol{p})$). Using the convention that the prices of products not offered are set to $+\infty$ or simply a value larger than $\max_{b \in \mathcal{B}} b$, we drop the dependence on S and re-write the optimization problem as $\max_{\boldsymbol{p} \in \mathbb{R}_+^n} R(\boldsymbol{p})$. Recall from Section 2 that the domain $\mathbb{R}_+^n$ is partitioned into K disjoint regions, with $\mathbb{R}_+^n = \mathcal{P}^1 \cup \mathcal{P}^2 \cup \cdots \cup \mathcal{P}^K$ and $\mathcal{P}^k \cap \mathcal{P}^{k'} = \varnothing$ if $k \neq k'$. Therefore, the optimization problem is equivalent to $\max_{k=1,\ldots,K} \max_{\boldsymbol{p} \in \mathcal{P}^k} R(\boldsymbol{p})$.

The decision problem is NP-complete in the strong sense even for the simplest case in which $\mathcal{B} = \{1, 2\}$ and $K = 1$ (Rusmevichientong et al. [30]). Therefore, we constrain the prices to be d-ordered, so the optimization problem that we wish to solve is

$$\max_{k=1,\ldots,K} \max_{\boldsymbol{p} \in \mathcal{P}^k \cap \mathscr{P}_d} R(\boldsymbol{p}) \ .$$

For a fixed d, we will show that the optimization problem admits a PTAS[4]. We first describe the PTAS for the single region case ($K = 1$) to facilitate exposition and highlight the key algorithmic techniques. We then extend the PTAS to the general ($K > 1$) case.

5.1. PTAS for a single region ($K = 1$)

We assume that d is fixed and the domain is not partitioned. So, our objective is to solve the following problem:

$$Z^* = \max_{\boldsymbol{p} \,\in\, \mathscr{P}_d} R(\boldsymbol{p}) \ ,$$

where $\mathscr{P}_d$ denotes the set of d-sorted prices in $\mathbb{R}_+^n$. The objective can be simplified as follows:

$$\begin{aligned} R(\boldsymbol{p}) &\stackrel{\text{def}}{=} \sum_{i=1}^n p_i \theta_i(\boldsymbol{p}) \;=\; \sum_{i=1}^n p_i \sum_{b \in \mathcal{B}} g_b \mathbb{P}_\lambda \left(i \mid \{a : p_a \le b\} \right) \\ &= \sum_{i=1}^n p_i \sum_{b \in \mathcal{B}} g_b \sum_{\boldsymbol{\sigma} \in \mathscr{S}_{n+1}} \lambda(\boldsymbol{\sigma}) \mathbb{1}\left[\boldsymbol{\sigma}, i, \{a : p_a \le b\}\right] \;=\; \sum_{(b, \boldsymbol{\sigma}) \in \mathcal{B} \times \mathscr{S}_{n+1}} w(b, \boldsymbol{\sigma}) \sum_{i=1}^n p_i \mathbb{1}\left[\boldsymbol{\sigma}, i, \{a : p_a \le b\}\right], \end{aligned}$$

where $w(b, \boldsymbol{\sigma}) \stackrel{\text{def}}{=} g_b \cdot \lambda(\boldsymbol{\sigma})$. We can interpret each pair $(b, \boldsymbol{\sigma})$ as a customer type, which comprises $w(b, \boldsymbol{\sigma})$ proportion of the population and has price threshold b and preference ordering $\boldsymbol{\sigma}$. The term $\sum_{i=1}^n p_i \mathbb{1}[\boldsymbol{\sigma}, i, \{a : p_a \le b\}]$ is the revenue from customer type $(b, \boldsymbol{\sigma})$ under price vector $\boldsymbol{p}$; note that $\{a \colon p_a \le b\}$ always includes the no-purchase option a_0 because $p_{a_0} = 0 \le b$ for all $b \ge 0$. Throughout this section, we assume without loss of generality that $R(\boldsymbol{p})$ can be computed in $O(1)$ operations for every price vector $\boldsymbol{p} \in \mathbb{R}_+^n$. Because $\mathcal{B}$ is finite, we assume, by scaling if necessary,

[4] The strong NP-completeness result eliminates the possibility of a fully polynomial-time approximation scheme (FPTAS) for general d. Thus, PTAS is the best that we can hope for.

that $\max_{b\in\mathcal{B}} b < 1$, so the price thresholds are strictly less than one. As a result of this scaling, setting the product price to 1 is equivalent to removing it from the offer set. The main result of this section is stated in the following theorem:

THEOREM 5.1 **(PTAS)**. *For any $\epsilon \in (0,1)$, there exists an algorithm $\mathcal{A}_\epsilon$ that generates a price vector $\boldsymbol{p}_\epsilon \in \mathscr{P}_d$ such that*

$$R(\boldsymbol{p}_\epsilon) \geq (1-\epsilon)Z^* \ ,$$

and the running time of algorithm $\mathcal{A}_\epsilon$ is $O\left(n^3 \times 4^{2d} \times k(\epsilon) \times n^{4(d+1)k(\epsilon)} \times \log \frac{1}{b_{\min}}\right)$, where

$$b_{\min} = \min_{b\in\mathcal{B}} b \qquad \text{and} \qquad k(\epsilon) = \left[\frac{4}{\epsilon} + \frac{4}{\epsilon}\ln\frac{1}{\epsilon}\right]^{1 \,+\, \frac{1}{\ln\frac{1}{\epsilon}}} .$$

We note that Aggarwal et al. [1] developed a PTAS for the special case of $d=0$, but their development does not extend to general d. Therefore, our development below involves new solution techniques. We provide a sketch of the proof of Theorem 5.1 and defer the details to Appendix C. The proof proceeds in three steps:

1. *Discretize the price domain.* We restrict attention to the discrete grid of prices $\mathsf{Dom}_\alpha \stackrel{\text{def}}{=} \{\alpha^s : s \in \mathbb{Z}_+\} = \{1, \alpha, \alpha^2, \dots\}$, where $\mathbb{Z}_+$ is the set of non-negative integers and $\alpha \in (0,1)$. We show that restriction to Dom_α results in an α-approximate solution i.e, $\max_{\boldsymbol{p}\in\mathscr{P}_{d,\alpha}} R(\boldsymbol{p}) \geq \alpha Z^*$, where $\mathscr{P}_{d,\alpha} \stackrel{\text{def}}{=} \{\boldsymbol{p}\in\mathscr{P}_d : p_i \in \mathsf{Dom}_\alpha \ \forall\ i\}$. See Lemma C.1 in Appendix C.
2. *Relax the revenue function.* The revenue function $R(\boldsymbol{p})$ is not directly amenable to optimization because the decision of a customer to purchase product i depends on the prices of all the other products. In order to limit this dependence, we relax the revenue function to allow a customer to purchase multiple products in each purchase instance. Specifically, given integer parameter k, define $R^{\alpha,k}(\boldsymbol{p}) \stackrel{\text{def}}{=} \sum_{(b,\boldsymbol{\sigma})} w(b,\boldsymbol{\sigma}) \sum_{i=1}^n \mathbb{1}\left[\boldsymbol{\sigma}, i, \left\{\ell : p_\ell \leq b \text{ and } p_\ell \leq p_i/\alpha^k\right\}\right]$, so that a customer previously purchasing product i may now also purchase lower-priced products that are at least k price levels apart. In other words, the purchased subset $\{j_1, j_2, \dots\}$ are such that $p_{j_\ell} \leq p_i \cdot \alpha^{(\ell-1)(k+1)}$ for any $\ell \geq 1$. With this relaxation, it is easy to see that a customer generating revenue of p_i previously now generates at least p_i and no more than $p_i \cdot (1 + \alpha^{k+1} + \alpha^{2(k+1)} + \cdots) = p_i/(1-\alpha^{k+1})$. It thus follows that $R(\boldsymbol{p}) \leq R^{\alpha,k}(\boldsymbol{p}) \leq R(\boldsymbol{p})/(1-\alpha^{k+1})$; see Lemma C.2 in Appendix C. Combining this with the approximation from the above step, we can show that if $\hat{\boldsymbol{p}}$ is the optimal solution to $\max_{\boldsymbol{p}\in\mathscr{P}_{d,\alpha}} R^{\alpha,k}(\boldsymbol{p})$, then $Z^* \geq R(\hat{\boldsymbol{p}}) \geq \alpha(1-\alpha^{k+1})Z^*$. See Proposition C.1 in Appendix C.
3. *Optimize the relaxed revenue function.* The above discretization of prices and relaxation of the revenue function limits the dependence of the decision to purchase product i to only products whose prices are no more than k levels larger than p_i. We exploit this fact to formulate the

optimization of the relaxed revenue function as a dynamic program. We elaborate this step further below.

The key step in the proof is to show that the relaxed revenue function can be optimized efficiently over $\mathscr{P}_{d,\alpha}$. Specifically, we can establish the following result:

PROPOSITION 5.1 **(DP for the relaxed problem)**. *The problem* $\max_{\boldsymbol{p} \in \mathscr{P}_{d,\alpha}} R^{\alpha,k}(\boldsymbol{p})$ *can be solved via a DP with a running time of* $O\left(n^3 \times 4^{2d} \times n^{4(k+1)(d+1)} \times \frac{\log 1/b_{\min}}{\log 1/\alpha}\right)$.

The details of the proof of Proposition 5.1 are deferred to the appendix; we provide an overview here. First, because it is never optimal to price a product below the minimum possible price threshold, we restrict product prices to $\{1, \alpha, \ldots, \alpha^H\}$; here, α^H is the largest price that is less than or equal to the smallest price threshold, i.e., $\alpha^H = \max\{\alpha^s : \alpha^s \le b_{\min}\}$ and $b_{\min} = \min_{b \in \mathcal{B}} b$. It follows from the definition that $H = O\left(\frac{\log(1/b_{\min})}{\log(1/\alpha)}\right)$. Given this, we represent each price vector $\boldsymbol{p}$ using the equivalent *subset representation*: $(A_H, A_{H-1}, \ldots, A_1, A_0)$, where $A_s \stackrel{\text{def}}{=} \{i : p_i = \alpha^s\}$ consists of products priced at α^s. The optimization problem now reduces to determining the optimal partitioning of the n products into subsets $(A_H, A_{H-1}, \ldots, A_1, A_0)$. To simplify notation, we let $A_{[\ell_1:\ell_2]}$ denote the tuple $(A_{\ell_1}, A_{\ell_1 - 1}, \ldots, A_{\ell_2})$ for any $\ell_1 \ge \ell_2$.

We determine the optimal partitioning in a sequential fashion as follows: for each $0 \le s \le H$, we fix the sets $A_{[H:s-k+1]}$ and consider the sub-problem of determining the maximum revenue $J_s^*(A_{[H:s-k+1]})$ – under the relaxed revenue function – that can be obtained from *only* the products with prices in $A_{[s:0]}$. Because every product is priced at least α^H, we can show that the optimal prices can be obtained by solving the optimization problem at $s = H$: $\max_{A_{[H:H-k+1]}} J_H^*(A_{[H:H-k+1]})$. Therefore, it is sufficient to show that we can determine $J_s^*(A_{[H:s-k+1]})$, for all possible s and $A_{[H:s-k+1]}$, and optimize over $A_{[H:H-k+1]}$ efficiently.

We determine $J_s^*(\cdot)$ by formulating it as the DP:

$$J_s^*(A_{[H:s-k+1]}) = \max_{A_{s-k} \in \mathcal{D}} \left(\alpha^s G_s(A_{[H:s-k]}) + J_{s-1}^*(A_{[H:s-k]})\right), \tag{12}$$

where $G_s(\cdot)$ is the number of customers who purchase products in A_s, and $\mathcal{D}$ appropriately restricts A_{s-k} so that $A_{[H:0]}$ is d-sorted.

To understand the recursion, let $W_s(A_{[H:0]})$ denote the revenue under the relaxed revenue function from the products in $A_{[s:0]}$. Then, we have, by definition, $J_s^*(A_{[H:s-k+1]}) \stackrel{\text{def}}{=} \max_{A_{[s-k:0]} \in \mathcal{D}'\left(A_{[H:s-k+1]}\right)} W_s(A_{[H:0]})$, where $\mathcal{D}'\left(A_{[H:s-k+1]}\right)$ denotes the domain of $A_{[s-k:0]}$ given $A_{[H:s-k+1]}$ such that $A_{[H:0]}$ is d-sorted. Now, the revenue $W_s(\cdot)$ can be decomposed into the revenue from the purchase of products in A_s and the revenue from products in $A_{[s-1:0]}$, i.e., we can write

$W_s(A_{[H:0]}) = \alpha^s G_s(A_{[H:0]}) + W_{s-1}(A_{[H:0]})$, where $G_s(A_{[H:0]})$ is the number of customers who purchase products in A_s. Under the relaxed revenue function, the customers who purchase products in A_i have a consideration set that is a subset of the $A_{[H:i-k]}$, products with prices less than or equal to k levels above the price of i. Therefore, $G_s(\cdot)$ will only depend on $A_{[H:s-k]}$. It now follows that

$$W_s(A_{[H:0]}) = \alpha^s G_s(A_{[H:s-k]}) + W_{s-1}(A_{[H:0]}) \ ,$$

which implies that

$$\begin{aligned}\max_{A_{[s-k:0]} \in \mathcal{D}'(A_{[H:s-k+1]})} W_s(A_{[H:0]}) &= \max_{A_{[s-k:0]} \in \mathcal{D}'(A_{[H:s-k+1]})} \left\{ \alpha^s G_s(A_{[H:s-k]}) + W_{s-1}(A_{[H:0]}) \right\} \\ &= \max_{A_{s-k} \in \mathcal{D}} \left\{ \alpha^s G_s(A_{[H:s-k]}) + \max_{A_{[s-k-1:0]} \in \mathcal{D}(A_{[H:s-k]})} W_{s-1}(A_{[H:0]}) \right\},\end{aligned}$$

where the last equality follows from decomposing the maximization over $A_{[s-k:0]}$ into our maximization over A_{s-k} and inner maximization over $A_{[s-k-1:0]}$ with the domains constrained appropriately. The DP recursion (12) now follows from the definition of J_s^*.

We then show that the DP can be solved efficiently by first arguing that the effective state space of each value function $J_s^*(\cdot)$ is small (see Lemma C.5 in Appendix C), consisting of $O\left(n^3 \times 4^{2d} \times n^{4k(d+1)}\right)$ distinct values. We then exploit the d-sorted price structure to show that the number of distinct subsets A_s is at most $O(n^{4(d+1)})$. It then follows from the DP recursion that given $J_{s-1}^*(\cdot)$, the value function $J_s^*(\cdot)$ can be computed in $O\left(n^3 \times 4^{2d} \times n^{4k(d+1)} \times n^{4(d+1)}\right) = O\left(n^3 \times 4^{2d} \times n^{4(k+1)(d+1)}\right)$, where the last term arises because the computation for each state requires maximization over $n^{4(d+1)}$ sets. Finally, because finding the optimal solution requires computing H value functions, the total complexity scales as $O(n^3 \times 4^{2d} \times n^{4(k+1)(d+1)} \times H) = O\left(n^3 \times 4^{2d} \times n^{4(k+1)(d+1)} \times \frac{\log(1/b_{\min})}{\log(1/\alpha)}\right)$. We argue that this term dominates the maximization over $A_{[H:H-k+1]}$, establishing the result.

Using the above result and choosing appropriate values for α and k, such that $\alpha(1-\alpha^{k+1}) \geq 1-\varepsilon$, establishes the result of Theorem 5.1.

5.2. Extension to multiple regions

We now extend the PTAS for $K=1$ to the general case with $K>1$ partitions. For the general case, we solve K sub-problems: $\max_{\boldsymbol{p} \in \mathcal{P}^k \cap \mathscr{P}_d} R(\boldsymbol{p})$ for each $1 \leq k \leq K$. Letting $\boldsymbol{p}_k^*$ denote the solution for the k^{th} sub-problem, we obtain the global optimum $\boldsymbol{p}^*$ as $\arg\max_{k=1,\ldots,K} R(\boldsymbol{p}_k^*)$. We solve each sub-problem approximately using the ideas described for the $K=1$ case. It is then clear that $\arg\max_{k=1,\ldots,K} R(\hat{\boldsymbol{p}}_k)$ provides the desired approximation to $\boldsymbol{p}^*$. Therefore, we focus this section on solving a sub-problem.

The complexity of solving $\max_{\boldsymbol{p}\in\mathcal{P}^k\cap\mathscr{P}_d} R(\boldsymbol{p})$ depends on the shape of $\mathcal{P}^k$. Arbitrary shapes can make the problem intractable. To avoid this, we focus on regions described by box constraints: $\mathcal{P}^k = \{\boldsymbol{p}\colon a_i \le p_i \le u_i \ \forall\ i\}$. We first discretize the price and maximize over the "rounded" domain $\mathcal{P}^k_\alpha \cap \mathscr{P}_{d,\alpha}$, where $\mathscr{P}_{d,\alpha}$ is as defined above and $\mathcal{P}^k_\alpha = \{\boldsymbol{p}\colon p_i \in \mathsf{Dom}_\alpha, \alpha^{L_i} \le p_i \le \alpha^{U_i} \ \forall\ i\}$, with $\alpha^{L_i} = \min\{\alpha^s\colon a_i \le \alpha^s\}$ and $\alpha^{U_i} = \max\{\alpha^s\colon \alpha^s \le u_i\}$. As shown in Appendix G, by considering the relaxed revenue function and adapting the DP technique from Proposition 5.1 in Section 5.1, we can obtain a PTAS for the optimization problem $\max_{\boldsymbol{p}\in\mathcal{P}^k_\alpha\cap\mathscr{P}_d} R(\boldsymbol{p})$. Then, under some technical conditions, we can show that the solution of the problem $\max_{\boldsymbol{p}\in\mathcal{P}^k_\alpha\cap\mathscr{P}_d} R(\boldsymbol{p})$ provides a good approximation to the original problem $\max_{\boldsymbol{p}\in\mathcal{P}^k\cap\mathscr{P}_d} R(\boldsymbol{p})$. The details are beyond the scope of this research and we do not pursue them here.

6. Numerical Study

We carried out two numerical studies to test our methods. The first study tests the *predictive* accuracy of our model on real-world sales transaction data from the IRI Academic Dataset (see Bronnenberg et. al. [10]). The second study tests the *decision* accuracy of our model on synthetic transaction data. The first test accomplishes two objectives: (a) it demonstrates the application of our methods in a real-world setting, and (b) it pits our method in a horse-race against the popular benchmark, the LC-MNL model. We find that our method obtains an average of 26% improvement over the benchmark on a 'chi-square' metric, which measures the relative error in predicting market shares. The second test demonstrates that our method can increase revenues by 11% compared to the LC-MNL benchmark by improving the joint assortment and pricing decision. We used synthetic data because we needed the ground-truth model to compare the "true" performance of the decisions. Next, we describe the details of the studies.

6.1. Predictive accuracy: Case study with the IRI Academic Dataset

The IRI Academic Dataset is a publicly available dataset containing real-world purchase transactions of consumer packaged goods (CPG) for chains of grocery and drug stores. The data consist of weekly sales transactions aggregated over all the customers. We focused on the transactions of three categories for the first two weeks in the year 2011: laundry, yogurt, and coffee. Each transaction contains the following information: the week and store of purchase, the universal product code (UPC) of the purchased product, quantity purchased, price paid, and an indicator of whether the product was on price or display promotion. The dataset for the first two weeks contained a total of approximately 220K, 544K, and 374K transactions from 1272 stores for laundry, yogurt, and coffee, respectively.

We processed the raw transactions to obtain products, prices, and offer sets, as described next. First, we dealt with data sparsity by aggregating the purchased items by vendors to obtain products. Each purchased item is identified in the dataset by its collapsed UPC, which is a 13-digit-long code with digits 4 to 8 (5 digits) denoting the vendor. There are a totals of 75, 90, and 290 vendors in the first two weeks of purchases for the yogurt, laundry, and coffee categories respectively. Because no-purchase sales are not observed, we made the assumption that the entire market is reasonably captured by all the stores and the vendors in the dataset. Further, the focus is on the revenues from the top 9 vendors, so the remaining vendors comprise the "rest of the market." Therefore, we treated the top 9 vendors as "products" and aggregated the remaining vendors into the "outside good" or the no-purchase option.

Then, we determined the context of offer set and prices for each purchase instance. Each combination of store and week results in an offer-set and price-vector combination. We inferred the offer set to be the union of all the products purchased during the particular week, at the particular store. We set the purchase price of a product equal to the weighted average of the prices of the different UPCs that comprise the product, where the weight of each price was equal to the corresponding observed sales at the particular store and week combination.

Our pre-processing resulted in a total of 2470 offer-set and price-vector combinations for each of the yogurt, laundry, and coffee categories.

6.1.1. Models fit. We compared the predictive accuracy of our model against the popular LC-MNL benchmark on the above dataset. We briefly describe how each of the models were fitted to the provided training data.

Benchmark LC-MNL model. The L-class LC-MNL model assumes that customers belong to one of L classes, for some non-negative integer L, and customers in class ℓ make choices according to a single-class MNL model with intercept vector $\boldsymbol{\mu}_\ell$ and price coefficient β_ℓ. A customer has a probability of α_ℓ of belonging to class ℓ, where $\alpha_\ell \geq 0$ for all ℓ and $\sum_{\ell=1}^{L} \alpha_\ell = 1$. With these assumptions, the probability that a customer in class ℓ purchases product c from offer-set and price combination $(S, \boldsymbol{p})$ is equal to $\exp(\mu_{\ell c} - \beta_\ell p_c) / \left(1 + \sum_{j \in S} \exp(\mu_{\ell j} - \beta_\ell p_j)\right)$. We estimated the model parameters by solving the following maximum-likelihood problem:

$$\max_{\boldsymbol{\mu}, \boldsymbol{\beta}, \boldsymbol{\alpha}} \sum_{t=1}^{T} \log \left(\sum_{\ell=1}^{L} \alpha_\ell \frac{\exp(\mu_{\ell c_t} - \beta_\ell p_{t,c_t})}{1 + \sum_{j \in S_t} \exp(\mu_{\ell j} - \beta_\ell p_{tj})} \right).$$

The above optimization problem is, in general, hard to solve (see Train [34]). We used the EM algorithm described in Train [35] to find a stationary point. We picked the LC-MNL as our benchmark because it can approximate any random utility choice model arbitrarily closely as the number of latent classes L increases [34].

Our nonparametric Joint Assortment and Price (JAP) model. We fitted our model using the EM algorithm described in Section 3.3. We discuss the implementation details of two core steps: (a) partitioning the training price vectors into K segments and (b) solving the M-step LP.

We clustered the training price vectors into K segments using the popular k-means algorithm and trained our nonparametric model separately for each segment. We set $K = 10$ using cross-validation. We observed that the predictive performance of the model was robust to the choice of K, as long as it was within a reasonable range. As mentioned above, we fitted a model to each cluster separately. For prediction, we mapped each test price vector to its closest cluster (as measured by the distance to the cluster centroid) and used the model trained on the corresponding cluster.

We solved the M-step LP by implementing a popular local search (LS) heuristic, instead of the DP discussed in Section 4, in order to emphasize the ease of practical implementation of our method. Our implementation demonstrates that even approximation implementations of the M-step LP yield accurate predictions. The LS heuristic starts with a random permutation and moves to the neighbor that results in the maximum improvement in the objective function. The moves are repeated until either a local optimum is reached, i.e., none of the moves result in an improvement in the objective function, or we hit a pre-specified limit on the number of moves. We used a limit of 10 moves in our experiments. Several natural definitions for the neighborhood of the permutations are possible. We define the neighborhood of a permutation as the collection of rankings that are obtained by swapping the positions of two products in the permutation at a time; the neighborhood therefore consists of $O(n^2)$ rankings. The LS heuristic with this definition of neighborhood has been shown to find good approximations to the optimal solution (see Ali and Meliă [3] and Schalekamp and van Zuylen [31]) for the problem of Kemeny optimization, which is a special case of M-step LP (as shown in Appendix B). The LS heuristic also has other desirable properties such as using fixed amount memory and ease of coding. The precise implementation details of the LS heuristic are present in Appendix D.

6.1.2. Experiments and results. We carried out a 2-fold cross-validation in which we randomly partitioned the offer-set and price combinations into two parts of roughly equal sizes, trained the models on one part and tested them on the other part, and repeated the process with the train and test sets interchanged. For each model, we measured the predictive accuracies in terms of two

% Improvements over LC-MNL Under Each Metric	Laundry	Yogurt	Coffee	Avg Across 3 Product Categories
MAPE	23.6%	12.2%	14.2%	16.67%
X2PE	37.2%	21.1%	21.9%	26.73%

Table 1 **Improvements in the predictive accuracy of our nonparametric JAP model against the LC-MNL benchmark. All the numbers are statistically significant at 5%.**

popular metrics, mean absolute percentage error (MAPE) and chi-square prediction error (X2PE), defined as follows: for each $\mathsf{model} \in \{\text{LC-MCL}, \mathsf{JAP}\}$, we compute

$$\text{MAPE}^{\mathsf{model}} = \frac{1}{\sum_{(S,\boldsymbol{p})\in\mathcal{T}} |S|} \sum_{(S,\boldsymbol{p})\in\mathcal{T}} \sum_{a\in S} \frac{\left|\hat{\theta}_a^{\mathsf{model}}(S,\boldsymbol{p}) - \theta_a^{\mathsf{actual}}(S,\boldsymbol{p})\right|}{\theta_a^{\mathsf{actual}}(S,\boldsymbol{p})},$$

where $\hat{\theta}_a^{\mathsf{model}}(S,\boldsymbol{p})$ is the *predicted* probability, under the fitted model, that a will be purchased when offered as part of the offer set S at price vector $\boldsymbol{p}$, while $\hat{\theta}_a^{\mathsf{actual}}(S,\boldsymbol{p})$ is the empirical probability computed from the observations in the test set. In the above expression, $\mathcal{T}$ denotes the collection of offer-set and price combinations in the test set.

Similarly, for each $\mathsf{model} \in \{\text{LC-MCL}, \mathsf{JAP}\}$, we compute

$$\text{X2PE}^{\mathsf{model}} = \frac{1}{\sum_{(S,\boldsymbol{p})\in\mathcal{T}} |S|} \sum_{(S,\boldsymbol{p})\in\mathcal{T}} \sum_{a\in S} \frac{\left(\hat{n}_a^{\mathsf{model}}(S,\boldsymbol{p}) - n_a^{\mathsf{actual}}(S,\boldsymbol{p})\right)^2}{0.5 + n_a^{\mathsf{actual}}(S,\boldsymbol{p})},$$

where $\hat{n}_a^{\mathsf{model}}(S,\boldsymbol{p})$ is the *predicted* number of purchases, under the fitted model, of product a when offered as part of the offer set S at price vector $\boldsymbol{p}$, and $n_a^{\mathsf{actual}}(S,\boldsymbol{p})$ is the actual number of observed purchases in the test set. We computed the predicted number of purchases by multiplying the predicted choice probability $\hat{\theta}_a^{\mathsf{model}}(S,\boldsymbol{p})$ with the number of customers who were offered $S,\boldsymbol{p}$ in the test set. The X2PE metric is similar to the popular chi-square measure of goodness-of-fit of the form $(O-E)^2/E$, where O refers to the observed value and E refers to the expected or predicted value.

Table 1 reports, for each product category, the *percentage improvements* in the MAPE and X2PE metrics under our nonparametric JAP model relative to the LC-MNL model, defined as

$$\frac{\text{MAPE}^{\text{LC-MNL}} - \text{MAPE}^{\mathsf{JAP}}}{\text{MAPE}^{\text{LC-MNL}}} \quad \text{and} \quad \frac{\text{X2PE}^{\text{LC-MNL}} - \text{X2PE}^{\mathsf{JAP}}}{\text{X2PE}^{\text{LC-MNL}}}.$$

Higher numbers are better. It is evident from the results that our method significantly outperforms the benchmark method across both metrics. In particular, we notice an average of 16% improvement under MAPE and 26% improvement for X2PE metrics.

6.2. Decision accuracy: Simulation study

We now describe the results from our simulation study, which compares the expected revenues obtained from the assortment and price decisions under our method and the benchmark LC-MNL model. The results demonstrate that on difficult ground-truth model instances, our model on average extracts 11.5% more revenues from the market than the benchmark method. The broad experimental setup is as follows:

1. Pick an instance of the LC-MNL model class as the ground-truth model.
2. Use the ground-truth model to generate transaction data for a collection of assortment and price combinations. The generated data are representative of data collected in practice.
3. Fit the simulated transaction data to our model and a benchmark LC-MNL (true) model.
4. Optimize both the fitted models to determine the joint assortment and price decisions.
5. Compare the ground-truth revenues from the two decisions to determine the average increase in revenues.

The above experimental procedure pits the decisions from our method to the decisions from the true model. We expect the true model to perform better *if sufficient data are available*. However, in practice, customers exhibit diverse and complex choice behaviors, and available data are limited. For such cases, it is no longer clear if fitting the true model will in fact yield the best performance. Our results demonstrate that for complex ground-truth models, our model provides better approximations than fitting the true model when available data are limited.

Ground-truth models generated. We randomly generated instances of the ground-truth model from the L-class LC-MNL model class. The number of products was $n = 9$ (excluding the no-purchase option). We considered $L = 5, 10, 15, 20$ latent classes. Product prices were chosen from the 21 levels in the set $\mathcal{P} = \{0.5, 0.525, 0.55, \ldots, 0.975, 1\}$, starting from 0.5 in increments of 0.025 up to 1. For each value of L, we randomly sampled 50 instances as follows:

1. The price of each product was sampled uniformly at random from the discrete set $\mathcal{P}$.
2. The mixing weight of each class was sampled uniformly at random from the interval $[0, 1]$ and then normalized so that the mixing weights sum to 1.
3. For each $1 \le \ell \le L$, the consideration set C_ℓ, consisting of up to five products, was sampled uniformly at random.
4. An extra class with the consideration set $\{1, 2, \ldots, n\} \setminus \cup_{k=1}^{K} C_k$, consisting of products not covered by any of the other consideration sets, if any, was added.
5. The following was done for each class:

(a) for each of the products in the consideration set corresponding to the class, an intercept term α was sampled uniformly at random from the interval $[-4, 1]$, a price coefficient β was sampled uniformly at random from the interval $[-3, -2]$, and the parameter v was set to $\exp(\alpha + \beta p)$, where p was the price of the product; and

(b) the parameter v of the remaining product was set to zero.

These models are computationally hard to estimate and optimize (see Désir and Goyal [12]).

Synthetic transaction data generated. From each instance of the ground-truth model described above, we generated synthetic transaction data as follows: we generated 30 price vectors by sampling the price of each product independently and uniformly at random from the discrete set $\mathcal{P}$. For each price vector, we generated 1000 offer sets by sampling subsets of sizes between two and eight uniformly at random. For each of the resulting 30,000 offer-set and price combinations, we randomly sampled a product choice according to the ground-truth model instance. The resulting data are of the form $(c_1, \boldsymbol{p}_1, S_1), (c_2, \boldsymbol{p}_2, S_2), \ldots, (c_T, \boldsymbol{p}_T, S_T)$, where $T = 30,000$. The above sampling mechanism mimics realistic settings in which prices change at a slower rate, but offer sets change at a faster rate because of stockouts, deliberate scarcity, or web page limitations.

Experiments conducted. For each ground-truth model instance, we fitted our nonparametric JAP and the benchmark LC-MNL models to the synthetic transaction data. We solved the following MLE problem to fit the benchmark LC-MNL model with k classes:

$$\max_{\boldsymbol{\mu},\boldsymbol{\beta},\boldsymbol{\alpha}} \sum_{t=1}^{T} \log\left(\sum_{\ell=1}^{L} \alpha_\ell \frac{\exp(\mu_{\ell,c_t} - \beta_{\ell,c_t} p_{t,c_t})}{1 + \sum_{j \in S_t} \exp(\mu_j - \beta_{\ell,j} p_{tj})}\right).$$

The above optimization problem is, in general, hard to solve (see Train [34]). We used the EM algorithm described in Train [35] to find a stationary point. We tuned the number of classes through cross-validation. To fit our model, we used the algorithm described in Section 3.3 and estimated parameters g and λ with a single partition.

For each of the fitted models, we computed estimates of the optimal joint assortment and price decision. We estimated the optimal decisions under the benchmark LC-MNL model and the non-parametric methods by solving the MILPs LC-MNL JOINT OPT and NONPARAMETRIC JOINT OPT, respectively. The MILPs are described in Appendix E. We solved the MILPs with a time limit of 40s using Gurobi Optimizer version 6.0.2 on a computer with a 3.5GHz Intel Core i5 processor, 16GB of RAM, and the Mac OSX Yosemite operating system. The MILPs may not be solved to optimality within the provided time limit, in which case we used the best solution obtained by Gurobi as the estimate of the optimal decision.

# latent classes (L)	5	10	15	20
% improvement	11.23%	12.31%	11.86%	10.53%

Table 2 **Improvements in the revenue under the decision computed from our nonparametric** JAP **model against the LC-MNL benchmark. All numbers are statistically significant at 5%.**

Results and discussion. For each number of latent classes $L \in \{5, 10, 15, 20\}$, we generated 50 ground-truth model instances. For each ground-truth model instance $q = 1, 2, \ldots, 50$, we computed the optimal assortment and price decisions $\left(S^{(q)}_{\mathsf{JAP}}, \boldsymbol{p}^{(q)}_{\mathsf{JAP}}\right)$ and $\left(S^{(q)}_{\text{LC-MNL}}, \boldsymbol{p}^{(q)}_{\text{LC-MNL}}\right)$ respectively under the JAP and LC-MNL models fitted to the transaction data generated from the q^{th} ground-truth model instance. We then evaluated the *percentage increase* in the (true) revenue extracted from using our JAP model vs. the benchmark LC-MNL model:

$$\mathsf{Diff}^{(q)} = \frac{R^{(q),\text{true}}\left(S^{(q)}_{\mathsf{JAP}}, \boldsymbol{p}^{(q)}_{\mathsf{JAP}}\right) - R^{(q),\text{true}}\left(S^{(q)}_{\text{LC-MNL}}, \boldsymbol{p}^{(q)}_{\text{LC-MNL}}\right)}{R^{(q),\text{true}}\left(S^{(q)}_{\text{LC-MNL}}, \boldsymbol{p}^{(q)}_{\text{LC-MNL}}\right)}$$

where $R^{(q),\text{true}}(\cdot,\cdot)$ denotes the true revenue function associated with the q^{th} instance.

Table 2 reports, for each number of latent classes L, the average percentage increase in the revenues extracted from the decision under our model relative to the decision under the fitted LC-MNL model; that is, $\frac{1}{50}\sum_{q=1}^{50} \mathsf{Diff}^{(q)}$. The results illustrate that, on average, our method can extract 11.5% more revenues from the market than the benchmark method.

Note that our experiments pit our model against the true model. We attribute the poor performance of the benchmark to the presence of different consideration sets for different customer segments. The training data are not sufficient for the estimation procedure to drive the parameter v to zero for products not in the consideration set. Our model, on the other hand, is designed to be flexible to capture complex choice patterns.

7. Conclusions

Motivated by the inflexibility of existing models to capture the joint effect of assortments and prices, we proposed a tractable, nonparametric joint assortment and price choice model. Our approach is data driven, makes few structural assumptions, and is designed to improve the accuracy of revenue predictions. Surprisingly, the model also allows for tractable estimation *and* tractable optimization. The key technical contribution of our work is the identification of classes of pricing structures of increasing complexity. We then related the complexity of the pricing structure with the computational burden of carrying out estimation and optimization. Our characterization allows us to establish theoretical guarantees for our estimation algorithm and design a PTAS for the joint assortment and price optimization problem.

Our work opens the door for many exciting future research directions. The core of our model is based on a two-stage choice process, one in which a customer first forms a consideration set of relevant products and then chooses from the consideration set. Existing work in marketing provides empirical support for such a two-stage process in which customers adopt screening heuristics to form consideration sets. Our work has shown the potential gains in predictive accuracy that can be obtained from the two-stage choice models. Exploring the additional flexibility afforded by consideration sets to obtain tractable, nonparametric models is an exciting future direction.

Another key aspect of our work is our ability to provide guarantees for solving the M-step LP. As discussed in Section 4, the M-step LP generalizes the popular Kemeny optimization problem. While it has been shown that the Kemeny optimization problem is NP-hard, very little work has been done on understanding the source of complexity to find tractable sub-problems. The d-sorted price characterization we obtain is one of the few general structures that has allowed for isolation of the source of complexity to arrive at algorithms with provable guarantees. Further exploration of the d-sorted price structures can allow us to obtain principled heuristics that have so far remained unexplored for the very important Kemeny optimization problem. Finally, it is surprising that the d-sorted pricing structures also allow us to design a PTAS for the joint assortment and price optimization problem. Taking the key intuitions behind the PTAS to design scalable optimization algorithms for practical-sized problems can have a huge practical impact.

Acknowledgement

The authors thank the Department Editor Yinyu Ye, the Associate Editor, and the three anonymous referees for their detailed and thoughtful comments, which have greatly improved the paper. They also thank Huseyin Topaloglu and Yayi Zou for pointing out a technical issue in one of the proofs in a previous version of the manuscript, which has led to a cleaner proof for the result that the joint assortment and price decision problem admits a PTAS. This research is supported in part by the National Science Foundation through grants CMMI-1454310, CMMI-1157569, and CMMI-1433396.

References

[1] G. Aggarwal, T. Feder, R. Motwani, and A. Zhu. Algorithms for multi-product pricing. In *Automata, Languages and Programming*, pages 72–83. Springer, 2004.

[2] J. W. Alba and A. Chattopadhyay. Effects of context and post-category cues on recall of competing brands. *Journal of Marketing Research*, 22(3):340–349, 1985.

[3] A. Ali and M. Meilă. Experiments with Kemeny ranking: What works when? *Mathematical Social Sciences*, 64:28–40, 2012.

[4] A. Alptekinoglu and J. H. Semple. The exponomial choice model. *Cox School of Business Working Paper*, 2013.

[5] P. P. Belobaba and C. Hopperstad. Boeing/MIT simulation study: PODS results update. In *1999 AGIFORS Reservations and Yield Management Study Group Symposium, April*, pages 27–30, 1999.

[6] J. A. Belonax and R. A. Mittlestaedt. Evoked set size as a function of number of choice criterion and information variability. In H. K. Hunt, editor, *Advances in consumer research*, volume 5, pages 48–51. Association for Consumer Research, Ann Arbor, MI, 1978.

[7] M. Ben-Akiva, D. McFadden, T. Gärling, D. Gopinath, J. Walker, D. Bolduc, A. Börsch-Supan, P. Delquié, O. Larichev, T. Morikawa, A. Polydoropoulou, and V. Rao. Extended framework for modeling choice behavior. *Marketing Letters*, 10(3):187–203, 1999.

[8] J. Blanchet, G. Gallego, and V. Goyal. A Markov chain approximation to choice modeling. *Columbia IEOR Working Paper*, 2013.

[9] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, Cambridge, UK, 2004.

[10] B. Bronnenberg, M. Kruger, and C. Mela. The IRI academic dataset. *Management Science*, 27(4):745–748, 2008.

[11] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[12] A. Désir and V. Goyal. Near-optimal algorithms for capacity constrained assortment optimization. *Available at SSRN 2543309*, 2014.

[13] L. Dong, P. Kouvelis, and Z. Tian. Dynamic pricing and inventory control of substitute products. *Manufacturing & Service Operations Management*, 11(2):317–339, 2009.

[14] P. Ebbes, D. Papies, and H. J. Van Heerde. The sense and non-sense of holdout sample validation in the presence of endogeneity. *Marketing Science*, 30(6):1115–1122, 2011.

[15] V. Farias, S. Jagabathula, and D. Shah. A new approach to modeling choice with limited data. *Management Science*, 59(2):305–322, 2013.

[16] G. Gallego and R. Wang. Multiproduct price optimization and competition under the nested logit model with product-differentiated price sensitivities. *Operations Research*, 62(2):450–461, 2014.

[17] T. J. Gilbride and G. M. Allenby. A choice model with conjunctive, disjunctive, and compensatory screening rules. *Marketing Science*, 23(3):391–406, 2004.

[18] A. Haensel and G. Koole. Estimating unconstrained demand rate functions using customer choice sets. *Journal of Revenue and Pricing Management*, 10(5):438–454, 2011.

[19] W. Hanson and K. Martin. Optimizing multinomial logit profit functions. *Management Science*, 42(7):992–1003, 1996.

[20] J. R. Hauser. Consideration-set heuristics. *Journal of Business Research*, 67(8):1688–1699, 2014.

[21] J. R. Hauser and B. Wernerfelt. An evaluation cost model of consideration sets. *Journal of Consumer Research*, 16(4):393–408, 1990.

[22] D. Honhon, S. Jonnalagedda, and X. Pan. Optimal algorithms for assortment selection under ranking-based consumer choice models. *Manufacturing and Service Operations Management*, 12(2):279–289, 2012.

[23] J. A. Howard and J. N. Sheth. *The theory of buyer behavior*. John Wiley, New York, NY, 1969.

[24] S. Jagabathula. Assortment optimization under general choice. *Available at SSRN*, 2014.

[25] H. Li and W. T. Huh. Pricing multiple products with the multinomial logit and nested logit models: Concavity and implications. *Manufacturing & Service Operations Management*, 13(4):549–563, 2011.

[26] S. Mahajan and G. J. van Ryzin. Retail inventories and consumer choice. In S. Tayur, R. Ganeshan, and M. Magazine, editors, *Quantitative models for supply chain management*, pages 491–551. Kluwer Academic Publishers, 1998.

[27] M. Meilă, K. Phadnis, A. Patterson, and J. A. Bilmes. Consensus ranking under the exponential model. *arXiv preprint arXiv:1206.5265*, 2012.

[28] T. L. Parkinson and M. Reilly. An information processing approach to evoked set formation. In W. L. Wilke, editor, *Advances in consumer research*, volume 6, pages 227–231. Association for Consumer Research, Ann Arbor, MI, 1979.

[29] R. M. Ratliff, V. Rao, C. P. Narayan, and K. Yellepeddi. A multi-flight recapture heuristic for estimating unconstrained demand from airline bookings. *Journal of Revenue and Pricing Management*, 7(2):153–171, 2008.

[30] P. Rusmevichientong, B. Van Roy, and P. W. Glynn. A nonparametric approach to multiproduct pricing. *Operations Research*, 54(1):82–98, 2006.

[31] Frans Schalekamp and Anke van Zuylen. Rank aggregation: Together we're strong. In *ALENEX*, pages 38–51. SIAM, 2009.

[32] I. Sher, J. T. Fox, K. I. Kim, and P. Bajari. Partial identification of heterogeneity in preference orderings over discrete choices. Technical report, National Bureau of Economic Research, 2011.

[33] Jing-Sheng Song and Zhengliang Xue. Demand management and inventory control for substitutable products. *Working paper*, 2007.

[34] K. Train. EM algorithms for nonparametric estimation of mixing distributions. *Journal of Choice Modelling*, 1(1):40–69, 2008.

[35] K. Train. *Discrete choice methods with simulation*. Cambridge University Press, Cambridge, UK, 2nd edition, 2009.

[36] G. L. Urban. Perceptor: A model for product positioning. *Management Science*, 21(8):858–871, 1975.

[37] Garrett van Ryzin and Gustavo Vulcano. An expectation-maximization algorithm to estimate a general class of nonparametric choice models. *NYU Stern Working Paper*, 2013.

[38] Garrett van Ryzin and Gustavo Vulcano. A market discovery algorithm to estimate a general class of nonparametric choice models. *Management Science*, 61(2):281–300, 2014.

[39] G. Vulcano, G. van Ryzin, and W. Chaar. OM practice—choice-based revenue management: An empirical study of estimation and optimization. *Manufacturing & Service Operations Management*, 12(3):371–392, 2010.

This page is intentionally blank. Proper e-companion title page, with INFORMS branding and exact metadata of the main paper, will be produced by the INFORMS office when the issue is being assembled.

Online Appendix:

A Nonparametric Joint Assortment and Price Choice Model

Srikanth Jagabathula
Stern School of Business, New York University, New York, NY 10012, sjagabat@stern.nyu.edu

Paat Rusmevichientong
Marshall School of Business, University of Southern California, Los Angeles, CA 90089, rusmevic@marshall.usc.edu

Appendix A: Proofs for Section 3

A.1. Proof of Lemma 3.1

Letting B_t denote the latent threshold observed for customer t, the *complete data log-likelihood* function is given by

$$\mathcal{L}_C \stackrel{\text{def}}{=} \sum_t \log\left(g_{B_t} \mathbb{P}_\lambda\left(c_t | S_t^{B_t}\right)\right) = \sum_t \log g_{B_t} + \sum_t \log \mathbb{P}_\lambda(c_t | S_t^{B_t}) \ ,$$

where for any $b \in \mathcal{B}$, $S_t^b = \{a \in S_t \cup \{a_0\} : p_{ta} \leq b\}$. We simplify the above expression by re-arranging and collecting terms as follows. We can simplify the first sum by collecting the terms corresponding to the same threshold level b together. Letting m_b denote the number of customers in the data with threshold b, we can write $\sum_t \log g_{B_t} = \sum_{b \in \mathcal{B}} m_b \log g_b$. Similarly, we simplify the second sum by collection terms corresponding to the same choice and consideration set pair (a, A) with $a = c_t$ and $A = S_t^{B_t}$ together. For that, we define the collection of all possible tuples $(c_t, S_t^{B_t})$:

$$\mathcal{C} \stackrel{\text{def}}{=} \left\{(a, A) \colon a = c_t \text{ and } A = S_t^b \text{ for some } b \in \mathcal{B} \text{ and } 1 \leq t \leq T\right\}.$$

Further, let $\gamma_{a,A}$ denote the number of customers t with the choice and consideration set pair (a, A). With this, it follows that $\sum_t \log \mathbb{P}_\lambda(c_t | S_t^{B_t}) = \sum_{(a,A) \in \mathcal{C}} \gamma_{a,A} \log \mathbb{P}_\lambda(a|A)$. Putting everything together, we get

$$\mathcal{L}_C = \sum_{b \in \mathcal{B}} m_b \log g_b + \sum_{(a,A) \in \mathcal{C}} \gamma_{a,A} \log \mathbb{P}_\lambda(a|A),$$

which is the desired result.

A.2. Proof of Lemma 3.2

We first compute the probability $h_t(b)$ that customer t has latent threshold b conditioned on the observation $o_t = (c_t, S_t, \boldsymbol{p}_t)$ and model parameters $\hat{g}$ and $\hat{\lambda}$. If B_t denotes the latent threshold of customer t, then we can write

$$\begin{aligned} h_t(b) \stackrel{\text{def}}{=} \mathbb{E}[\mathbb{1}_{\{B_t = b\}} | o_t, \hat{g}, \hat{\lambda}] = \Pr_{\hat{g},\hat{\lambda}}\left(B_t = b | o_t\right) &= \frac{\Pr_{\hat{g},\hat{\lambda}}(o_t | B_t = b) \Pr_{\hat{g},\hat{\lambda}}(B_t = b)}{\sum_{b' \in \mathcal{B}} \Pr_{\hat{g},\hat{\lambda}}(o_t | B_t = b') \Pr_{\hat{g},\hat{\lambda}}(B_t = b')} \\ &= \frac{\mathbb{P}_{\hat{\lambda}}(c_t | S_t^b) \hat{g}_b}{\sum_{b' \in \mathcal{B}} \mathbb{P}_{\hat{\lambda}}(c_t | S_t^{b'}) \hat{g}_{b'}}. \end{aligned}$$

Furthermore, we have that the expected number of customers with latent threshold b is given by $m_b = \sum_t \mathbb{1}_{\{B_t = b\}}$ and the expected number of customers with choice, consideration-set tuple (a, A) is given by $\gamma_{a,A} = \sum_t \sum_{b \in \mathcal{B}} \mathbb{1}_{\{B_t = b\}} \mathbb{1}_{\left\{c_t = a, S_t^b = A\right\}}$. With this we can write

$$\hat{m}_b = \mathbb{E}[m_b | \hat{g}, \hat{\lambda}] = \sum_{t=1}^T \mathbb{E}[\mathbb{1}_{\{B_t = b\}} | \hat{g}, \hat{\lambda}] = \sum_{t=1}^T h_t(b)$$

and

$$\hat{\gamma}_{a,A} = \mathbb{E}[\gamma_{a,A}|\hat{g},\hat{\lambda}] = \sum_{t=1}^{T}\sum_{b\in\mathcal{B}} \mathbb{E}[\mathbb{1}_{\{B_t=b\}}|\hat{g},\hat{\lambda}]\mathbb{1}_{\left\{c_t=a,S_t^b=A\right\}} = \sum_{t=1}^{T}\sum_{b\in\mathcal{B}} h_t(b)\mathbb{1}_{\left\{c_t=a,S_t^b=A\right\}}$$

Since the complete log-likelihood function is given by $\mathcal{L}_C = \sum_{b\in\mathcal{B}} m_b \log g_b + \sum_{(a,A)\in\mathcal{C}} \gamma_{a,A}\mathbb{P}_\lambda(a|A)$, the result of the lemma follows from the above expressions for $\mathbb{E}[m_b|\hat{g},\hat{\lambda}]$ and $\mathbb{E}[\gamma_{a,A}|\hat{g},\hat{\lambda}]$.

A.3. Proof of Proposition 3.1

In order to prove the result of the proposition, we need the following lemma.

LEMMA A.1. *Suppose we are given $\hat{\boldsymbol{y}} \in Q_\mathcal{C}$ such that $\hat{\boldsymbol{y}} > 0$. Then, for any $\boldsymbol{y} = \alpha\hat{\boldsymbol{y}} + (1-\alpha)\boldsymbol{x}$ with $0 < \alpha \le 1$ and $\boldsymbol{x} \in Q_\mathcal{C}$, we must have*

$$f(\boldsymbol{y}) \le f(\hat{\boldsymbol{y}}) + (1-\alpha)\left[-T + \sum_{(a,A)\in\mathcal{C}} c_{a,A}x_{a,A}\right].$$

where $c_{a,A} \overset{def}{=} \hat{\gamma}_{a,A}/\hat{y}_{a,A}$. Further, if $\boldsymbol{y}, \hat{\boldsymbol{y}} \ge \eta > 0$, then we must have

$$f(\boldsymbol{y}) \ge f(\hat{\boldsymbol{y}}) + (1-\alpha)\left[-T + \sum_{(a,A)\in\mathcal{C}} c_{a,A}x_{a,A}\right] - 2((1-\alpha)/\eta)^2T^2$$

where $\boldsymbol{y} = \alpha\hat{\boldsymbol{y}} + (1-\alpha)\boldsymbol{x}$.

Proof. We first establish the upper bound. For that, we use the subgradient inequality for $\log(\cdot)$. Specifically, since $\log(\cdot)$ is a strictly concave function, we have the subgradient inequality at $z_0 \in \mathbb{R}_+$

$$\log z \le (\log z_0 - 1) + z/z_0$$

for any $z \in \mathbb{R}_+$ with equality occurring if and only if $z = z_0$. We assume that $\hat{y}_{a,A} > 0$ for all $(a,A) \in \mathcal{C}$. Thus, we can apply the above inequality with $z_0 = \hat{y}_{a,A}$ to obtain

$$\log y_{a,A} \le (\log \hat{y}_{a,A} - 1) + y_{a,A}/\hat{y}_{a,A} \text{ for any } (a,A) \in \mathcal{C}.$$

Since $\hat{\gamma}_{a,A} > 0$ for all $(a,A) \in \mathcal{C}$, we can now write

$$\sum_{(a,A)\in\mathcal{C}} \hat{\gamma}_{a,A}\log y_{a,A} \le \sum_{(a,A)\in\mathcal{C}} \hat{\gamma}_{a,A}\log\hat{y}_{a,A} - \sum_{(a,A)\in\mathcal{C}} \hat{\gamma}_{a,A} + \sum_{(a,A)\in\mathcal{C}} \frac{\hat{\gamma}_{a,A}}{\hat{y}_{a,A}} y_{a,A},$$

where we require that $y_{a,A} > 0$ for all $(a,A) \in \mathcal{A}$. Since $\hat{\gamma}_{a,A}$ is the expected number of customers with consideration set A purchasing product a, we must have that $\sum_{(a,A)\in\mathcal{C}} \hat{\gamma}_{a,A} = T$, the total number of customers. Now let $\boldsymbol{y} = \alpha\hat{\boldsymbol{y}} + (1-\alpha)\boldsymbol{x}$ for some $\boldsymbol{x} \in Q_\mathcal{C}$ and $\alpha \in (0,1]$. Since $\hat{\boldsymbol{y}} > 0$, $\boldsymbol{x} \ge 0$, and $\alpha > 0$, it follows that $\boldsymbol{y} > 0$. Hence, we can use the above upper bound inequality with $\boldsymbol{y} = \alpha\hat{\boldsymbol{y}} + (1-\alpha)\boldsymbol{x}$ and letting $c_{a,A}$ denote $\hat{\gamma}_{a,A}/\hat{y}_{a,A}$ to get

$$\begin{aligned}
\sum_{(a,A)\in\mathcal{C}} \hat{\gamma}_{a,A}\log y_{a,A} &\le \sum_{(a,A)\in\mathcal{C}} \hat{\gamma}_{a,A}\log\hat{y}_{a,A} - T + \sum_{(a,A)\in\mathcal{C}} c_{a,A}(\alpha\hat{y}_{a,A} + (1-\alpha)x_{a,A}) \\
&= \sum_{(a,A)\in\mathcal{C}} \hat{\gamma}_{a,A}\log\hat{y}_{a,A} - T + \alpha\sum_{(a,A)\in\mathcal{C}} \hat{\gamma}_{a,A} + (1-\alpha)\sum_{(a,A)\in\mathcal{C}} c_{a,A}x_{a,A} \\
&= \sum_{(a,A)\in\mathcal{C}} \hat{\gamma}_{a,A}\log\hat{y}_{a,A} + (1-\alpha)\left[-T + \sum_{(a,A)\in\mathcal{C}} c_{a,A}x_{a,A}\right],
\end{aligned}$$

where the first equality follows from $c_{a,A}\hat{y}_{a,A} = \hat{\gamma}_{a,A}$ and the second equality follows from $\sum_{(a,A)\in\mathcal{C}} \hat{\gamma}_{a,A} = T$. The upper bound result now follows noting the definition of $f(\cdot)$ that $f(\boldsymbol{y}) = \sum_{(a,A)\in\mathcal{C}} \hat{\gamma}_{a,A} \log y_{a,A}$.

We now prove the lower bound result. In order to obtain the lower bound, we consider the second-order Taylor expansion around $z = z_0 > 0$ for $\log(\cdot)$ to write

$$\log z = \log z_0 + \frac{z - z_0}{z_0} - \frac{(z - z_0)^2}{\zeta^2}$$

for some ζ between z_0 and z. We focus on the domain bounded away from zero. Specifically, let $\eta > 0$ be the lower bound on z, z_0, so that have $-1/\zeta^2 \geq -1/\eta^2$ for any ζ between z and z_0. We must then have

$$\log z \geq \log z_0 - 1 + z/z_0 - \frac{(z - z_0)^2}{\eta^2}.$$

Now consider $\boldsymbol{y} \geq \eta$. Further, let $\boldsymbol{x} \in Q_{\mathcal{C}}$ and $\alpha \in [0,1]$ be such that $\boldsymbol{y} = \alpha\hat{y} + (1-\alpha)\boldsymbol{x}$; such $\boldsymbol{x}$ and α can always be found since $\mathcal{Q}_C$ is convex. Since $\hat{\boldsymbol{y}} \geq \eta$, applying the above inequality at $z = \boldsymbol{y}_{a,A}$ and $z_0 = \hat{y}_{a,A}$ for some $(a,A) \in \mathcal{C}$, we get

$$\log y_{a,A} \geq \log \hat{y}_{a,A} + (1-\alpha)\left[-1 + \frac{x_{a,A}}{\hat{y}_{a,A}}\right] - ((1-\alpha)/\eta)^2 (\hat{y}_{a,A} - x_{a,A})^2.$$

Summing the above inequality over $(a,A) \in \mathcal{C}$ and using arguments similar to above, we get

$$\sum_{(a,A)\in\mathcal{C}} \hat{\gamma}_{a,A} \log y_{a,A} \geq \sum_{(a,A)\in\mathcal{C}} \hat{\gamma}_{a,A} \log \hat{y}_{a,A} + (1-\alpha)\left[-T + \sum_{(a,A)\in\mathcal{C}} c_{a,A} x_{a,A}\right] - ((1-\alpha)/\eta)^2 \sum_{(a,A)\in\mathcal{C}} (\hat{y}_{a,A} - x_{a,A})^2$$

Now note that

$$\sum_{(a,A)\in\mathcal{C}} (\hat{y}_{a,A} - x_{a,A})^2 \leq \sum_{(a,A)\in\mathcal{C}} (\hat{y}_{a,A}^2 + x_{a,A}^2) \leq \left(\sum_{(a,A)\in\mathcal{C}} \hat{y}_{a,A}\right)^2 + \left(\sum_{(a,A)\in\mathcal{C}} x_{a,A}\right)^2 = 2T^2,$$

where the first and second inequalities follow from the fact that $\hat{y}_{a,A}, x_{a,A} \geq 0$ for all $(a,A) \in \mathcal{C}$ and the last equality follows from the fact that $\sum_{(a,A)\in\mathcal{C}} \hat{y}_{a,A} = T$ and $\sum_{(a,A)\in\mathcal{C}} x_{a,A} = T$, the total number of customers, because $\hat{\boldsymbol{y}}, \boldsymbol{x} \in Q$. We can now write

$$\sum_{(a,A)\in\mathcal{C}} \hat{\gamma}_{a,A} \log y_{a,A} \geq \sum_{(a,A)\in\mathcal{C}} \hat{\gamma}_{a,A} \log \hat{y}_{a,A} + (1-\alpha)\left[-T + \sum_{(a,A)\in\mathcal{C}} c_{a,A} x_{a,A}\right] - 2((1-\alpha)/\eta)^2 T^2.$$

This establishes the result of the lemma. ■

We now use the result of the lemma to establish the result of the proposition. For that first, suppose $v^* \leq 0$, where $v^* \stackrel{\text{def}}{=} -T + \max_{\boldsymbol{x}\in Q_{\mathcal{C}}} \sum_{(a,A)\in\mathcal{C}} c_{a,A} x_{a,A}$. Since any $\boldsymbol{y} \in Q_{\mathcal{C}}$ can be written as $\boldsymbol{y} = \alpha\hat{\boldsymbol{y}} + (1-\alpha)\boldsymbol{x}$, for some $\boldsymbol{x} \in Q_{\mathcal{C}}$, $\alpha \in [0,1]$, it follows from the upper bound in Lemma A.1 that

$$f(\boldsymbol{y}) \leq f(\hat{\boldsymbol{y}}) + (1-\alpha)\left[-T + \sum_{(a,A)\in\mathcal{C}} c_{a,A} x_{a,A}\right] \leq f(\hat{y}) + (1-\alpha)v^* \leq f(\hat{\boldsymbol{y}}).$$

Thus, $\hat{\boldsymbol{y}}$ is an optimal solution to $\max_{\boldsymbol{y}\in Q_{\mathcal{C}}} f(\boldsymbol{y})$ and we cannot finding an improving solution.

Now suppose $v^* > 0$ and $\boldsymbol{x}^* \in Q_{\mathcal{C}}$ is such that $v^* = -T + \sum_{(a,A)\in\mathcal{C}} c_{a,A} x^*_{a,A}$. We argue that there exists an $\alpha \in (0,1]$ such that $f(\boldsymbol{y}) > f(\hat{\boldsymbol{y}})$ for $\boldsymbol{y} = \alpha\hat{\boldsymbol{y}} + (1-\alpha)\boldsymbol{x}^*$.

To see this, we focus on $\alpha \in (\eta/\hat{y}_{\min}, 1]$, where $\hat{y}_{\min} = \min_{(a,A)\in\mathcal{C}} \hat{y}_{a,A}$ and $\eta < \hat{y}_{\min}$. Then, it follows that $\boldsymbol{y} \geq \alpha\hat{\boldsymbol{y}} \geq \alpha\hat{y}_{\min} \geq \eta$. Thus, we can invoke the lower bound result from Lemma A.1 to write

$$f(\boldsymbol{y}) \geq f(\hat{\boldsymbol{y}}) + (1-\alpha)\left[-T + \sum_{(a,A)\in\mathcal{C}} c_{a,A}x_{a,A}\right] - 2((1-\alpha)/\eta)^2T^2 = f(\hat{\boldsymbol{y}}) + (1-\alpha)\left[v^* - 2T^2\frac{1-\alpha}{\eta^2}\right].$$

Now, in order to show that $\boldsymbol{y}$ strictly improve on $\hat{\boldsymbol{y}}$, it is sufficient exhibit an $\alpha \in (\eta/\hat{y}_{\min}, 1)$ such that $v^* > 2T^2(1-\alpha)/\eta^2$. Equivalently, we must choose $\alpha > 1 - 0.5(v^*\eta^2/T^2)$. Combining this constraint on α with $\alpha \in (\eta/\hat{y}_{\min}, 1]$, we must have

$$\max\left\{\frac{\eta}{\hat{y}_{\min}}, 1 - \frac{v^*\eta^2}{2T^2}\right\} < \alpha < 1.$$

It follows from the above inequalities that if $\eta < \hat{y}_{\min}$, then the solution space defined by the above set of inequalities is non-empty. It thus follows that whenever $v^* > 0$, there exists an $\alpha \in (0,1)$ that strictly improves the solution over $\hat{\boldsymbol{y}}$.

An improving α can be found by solving the following one-dimensional search problem

$$\arg\max_{0\leq\alpha\leq 1} f(\alpha\hat{\boldsymbol{y}} + (1-\alpha)\boldsymbol{x}^*) = \arg\max_{0\leq\alpha\leq 1} \sum_{(a,A)\in\mathcal{C}} c_{a,A}\log\left(\alpha\left(\hat{y}_{a,A} - x^*_{a,A}\right) + x^*_{a,A}\right).$$

Due to the global concavity of $\log(\cdot)$ and the fact that a linear combination with non-negative cofficients of a collection of concave functions is concave, it follows that the above optimization problem is concave maximization over a single variable. Hence, it can be carried out efficiently. The result of the proposition now follows. ■

A.4. EM algorithm for model parameter estimation

Input $\mathsf{Data} = \{(c_1, \boldsymbol{p}_1), (c_2, \boldsymbol{p}_2, \ldots, (c_T, \boldsymbol{p}_T))\}$; number of products n, finite threshold space $\mathcal{B}$.

Initialization: Construct the collection of tuples $\mathcal{C} = \{(a, A) \colon a \in A, A = \{j \colon p_{tj} \leq b\}$ for some $b \in \mathcal{B}\}$. Determine initial estimates $\boldsymbol{y}^{(0)}$ such that $\boldsymbol{y}^{(0)} \in Q_{\mathcal{C}}$ and $\boldsymbol{y}_{a,A} > 0$ for all $(a,A) \in \mathcal{C}$. Also, let $\left(g_b^{(0)} : b \in \mathcal{B}\right)$ denote an initial PMF over the thresholds such that $g_b^{(0)} > 0$ for all b.

EM iterations: For $k = 1, 2, \ldots$ do

E-step: Compute $\left(h_t^{(k)}(b) : b \in \mathcal{B}, t = 1, \ldots, T\right)$ and $\left(\gamma_{a,A}^{(k)} : (a,A) \in \mathcal{C}\right)$, where

$$h_t^{(k)}(b) = \frac{y_{c_t, S_t^b}\, g_b^{(k)}}{\sum_{q\in\mathcal{B}} y_{c_t, S_t^q}\, g_q^{(k)}} \qquad \text{and} \qquad \gamma_{a,A}^{(k)} = \sum_{t=1}^{T}\sum_{b\in\mathcal{B}} h_t^{(k)}(b)\mathbb{1}_{\{c_t = a, A = S_t^b\}},$$

where $S_t^b = \{a : p_{ta} \leq b\}$ for all t and b.

M-step: Compute $\left(g_b^{(k+1)} \ : \ b \in \mathcal{B}\right)$ and $\left(y_{a,A}^{(k+1)} \ : \ (a,A) \in \mathcal{C}\right)$, where

$$g_b^{(k+1)} = \frac{1}{T}\sum_{t=1}^{T} h_t(b)$$

$$\boldsymbol{x}^{(k+1)} = \arg\max_{\boldsymbol{x}\in Q} \sum_{(a,A)\in Q} \left(\hat{\gamma}_{a,A}/y_{a,A}^{(k)}\right) x_{a,A}$$

$$\boldsymbol{y}^{(k+1)} = (1-\varepsilon^*)\boldsymbol{y}^{(k)} + \varepsilon^*\boldsymbol{x}^{(k+1)}, \text{ where } \varepsilon^* = \arg\max_{0\leq\varepsilon\leq 1} \sum_{(a,A)\in\mathcal{C}} \hat{\gamma}_{a,A}\log\left((1-\varepsilon)\boldsymbol{y}^{(k)} + \varepsilon\boldsymbol{x}^{(k+1)}\right)$$

Until: Stopping criteria are met.

Output. Sequence of estimates $\langle g^{(k)}, \boldsymbol{y}^{(k)} : k = 0, 1, 2, \dots \rangle$

Appendix B: Proofs for Section 4

B.1. Proof of Proposition 4.1

We show that the following optimization problem is NP-hard to solve:

$$\arg\max_{\boldsymbol{\sigma}} \sum_{(a,A)\in\mathcal{C}} c_{a,A} \mathbb{1}[\boldsymbol{\sigma}, a, A].$$

For that, we obtain a reduction from the Kemeny optimization problem – a known NP-hard problem for $n \geq 4$; see Ali and Meliă [1].

The Kemeny optimization problem is a classical rank aggregation problem. The setup is as follows. We are given a collection of total orderings/rankings over n items, and our goal is find a single ranking that minimizes the average 'distance' from all the given rankings. This problem arises in the borader context of 'rank aggregation', in which the goal is to find the most consistent ranking, given a multitude of preferences. The distance measure between rankings that is used in the context of Kemeny optimization is the so-called Kendall-tau distance that specifies that the distance between two rankings $\boldsymbol{\sigma}$ and $\boldsymbol{\pi}$ is equal to the number of pairs $a \neq a'$ for which the relative prference under $\boldsymbol{\sigma}$ and $\boldsymbol{\pi}$ are different. More formally, we define the distance

$$d(\boldsymbol{\sigma}, \boldsymbol{\pi}) = \sum_{a \neq a'} \mathbb{1}[(\sigma^{-1}(a) - \sigma^{-1}(a'))(\pi^{-1}(a) - \pi^{-1}(a')) < 0].$$

With this defition of the distance, now suppose we are given a collection of K rankings $\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \dots, \boldsymbol{\pi}_K$. The Kemeny optimization problem then is to find the best ranking $\boldsymbol{\sigma}^*$ defined as

$$\boldsymbol{\sigma}^* = \arg\max_{\boldsymbol{\sigma}} \sum_{k=1}^{K} d(\boldsymbol{\sigma}, \boldsymbol{\pi}_k).$$

To see how the Kemeny optimization problem reduces to our optimization problem of interest, define the collection of tuples $\mathcal{C} = \{(a, \{a, a'\}) : a \neq a', a, a' \in \mathcal{N}\}$. We can then write

$$\begin{aligned}
\sum_{k=1}^{K} d(\boldsymbol{\sigma}, \boldsymbol{\pi}) &= \sum_{k=1}^{K} \sum_{a \neq a'} \mathbb{1}[(\sigma^{-1}(a) - \sigma^{-1}(a'))(\pi_k^{-1}(a) - \pi_k^{-1}(a')) < 0] \\
&= \sum_{a \neq a'} \sum_{k=1}^{K} \mathbb{1}[(\sigma^{-1}(a) - \sigma^{-1}(a'))(\pi_k^{-1}(a) - \pi_k^{-1}(a')) < 0] \\
&= \sum_{a \neq a'} \sum_{k=1}^{K} \left(\mathbb{1}[\boldsymbol{\sigma}, a, \{a, a'\}] \mathbb{1}[\pi_k^{-1}(a') < \pi_k^{-1}(a)] + \mathbb{1}[\boldsymbol{\sigma}, a', \{a, a'\}] \mathbb{1}[\pi_k^{-1}(a) < \pi_k^{-1}(a')] \right) \\
&= \sum_{(a,\{a,a'\})\in\mathcal{C}} \mathbb{1}[\boldsymbol{\sigma}, a, \{a, a'\}] \left[\sum_{k=1}^{K} \mathbb{1}[\pi_k^{-1}(a') < \pi_k^{-1}(a)] \right] \\
&= \sum_{(a,A)\in\mathcal{C}} c_{a,A} \mathbb{1}[\boldsymbol{\sigma}, a, A],
\end{aligned}$$

where we define $c_{a,A}$ to be $\sum_{k=1}^{K} \mathbb{1}[\pi_k^{-1}(a') < \pi_k^{-1}(a)]$, the number of ranked lists $\boldsymbol{\pi}_k$ that prefer a' to a in subset $A = \{a, a'\}$. Thus, if we can solve our optimization problem efficiently, then we should be able to solve the Kemeny optimization problem efficiently, contradicting the fact that the Kemeny optimization problem is NP-hard.

This establishes the result of the Proposition. ■

B.2. Proof of Proposition 4.2

Our goal is to solve the following optimization problem

$$\arg\max_{\boldsymbol{\sigma}} \sum_{(a,A)\in\mathcal{C}} c_{a,A}\, \mathbb{1}[\boldsymbol{\sigma}, \boldsymbol{a}, \boldsymbol{A}] \tag{EC.1}$$

where $\mathcal{C} \subseteq \mathcal{C}_d$ for any given integer $0 \le d \le n$with the collection $\mathcal{C}_d$ is defined as

$$\mathcal{C}_d = \left\{(a, A) \colon a \in A, A = \{a' \colon p_{a'} \le b\} \text{ for some } b \in \mathcal{B} \text{ and } \boldsymbol{p} \in \mathscr{P}_d\right\}.$$

We solve the above optimization problem by formulating it as a Dynamic Program (DP) in which we construct the optimal preference list $\boldsymbol{\sigma}^*$ sequentially in the order of their preference, starting from the most preferred product. For that, let σ_r denote the product ranked at position r according to preference list $\boldsymbol{\sigma}$. The objective function in (EC.1) at $\boldsymbol{\sigma}$ can be written as

$$\sum_{(a,A)\in\mathcal{C}} c_{a,A}\, \mathbb{1}[\boldsymbol{\sigma}, \boldsymbol{a}, \boldsymbol{A}] = \sum_{r=0}^{n} \sum_{(\sigma_r, A)\in\mathcal{C}_d} c_{\sigma_r,A}\, \mathbb{1}[\boldsymbol{\sigma}, \sigma_r, A] = \sum_{r=0}^{n} \sum_{(\sigma_r,A)\in\mathcal{C} \colon A\cap\left\{\sigma_0,\sigma_1,\ldots,\sigma_{r-1}\right\}=\emptyset} c_{\sigma_r,A}\, \mathbb{1}[\boldsymbol{\sigma}, \sigma_r, A],$$

where the first equality follows from a straightforward rearrangement of the terms and the second equality follows from the fact that $\mathbb{1}[\boldsymbol{\sigma}, \sigma_r, A] = 0$ for any subset A that contains any of the products in the set $\{\sigma_0, \sigma_1, \ldots, \sigma_{r-1}\}$ that are preferred to σ_r. The above reformulation suggests the following DP formulation:

$$V_r(\bar{\sigma}_r) = \max_{\sigma_r \notin \bar{\sigma}_r} \left\{ \sum_{(\sigma_r,A)\in\mathcal{C} \colon A\cap\bar{\sigma}_r=\emptyset} c_{\sigma_r,A} + V_{r+1}(\bar{\sigma}_{r+1}) \right\},$$

with the boundary condition $V_{n+1}(\cdot) \equiv 0$ and $\bar{\sigma}_0 = \emptyset$ for all $\boldsymbol{\sigma}$ and $\bar{\sigma}_r \stackrel{\text{def}}{=} (\sigma_0, \sigma_1, \ldots, \sigma_{r-1})$ is the tuple consisting of the products ranked at the first $r-1$ positions. It can be seen that the optimal value to (EC.1) is given by $V_0(\bar{\sigma}_0)$ in the above DP. The optimal solution may be constructed through backward induction.

The dominating components in the computational complexity of the above DP are the cardinality of the state space and the number of choice,consideration set tuples. First note that for each value of $\bar{\sigma}_r$, computing the value of $V_r(\bar{\sigma}_r)$ requires us to search over all $\sigma_r \notin \bar{\sigma}_r$, assuming we are provided a look-up table for $V_{r+1}(\bar{\sigma}_{r+1})$ for each possible value of $\bar{\sigma}_{r+1}$; thus, this step has a worst-case computational complexity of $O(n)$. Furthermore, summing over all (σ_r, A) requires $O(|\mathcal{C}|)$ computations. Thus, for each value of $\bar{\sigma}_r$, the overall computational complexity is bounded above by $O(n\,|\mathcal{C}|)$.

Next we determine the number of distinct values for which we need to compute $V_r(\cdot)$. We show that when the choice, consideration-set combinations are restricted to belong to $\mathcal{C}_d$, we don't have to compute $V_r(\cdot)$

for all possible values of $\bar{\sigma}_r$. The reason is that there exists a *sufficient statistic* $\tau(\bar{\sigma}_r)$ such that $V_r(\bar{\sigma}_r)$ is completely determined by the sufficient statistic $\tau(\bar{\sigma}_r)$. An implication is that $V_r(\bar{\sigma}_r) = V_r(\bar{\sigma}'_r)$ whenever $\tau(\bar{\sigma}_r) = \tau(\bar{\sigma}'_r)$ even though $\bar{\sigma}_r \neq \bar{\sigma}'_r$. Therefore, we only need to compute $V_r(\cdot)$ for distinct values of $\tau(\bar{\sigma}_r)$. If $\tau(\bar{\sigma}_r)$ can take at most X_r values, it immediately follows that the computational complexity of computing $V_r(\bar{\sigma}_r)$ for all possible values of $\bar{\sigma}_r$ (assuming we have been provided a lookup table for $V_{r+1}(\bar{\sigma}_{r+1})$ for all possible values of $\bar{\sigma}_{r+1}$) is bounded above by $O(n\,|\mathcal{C}|\,X_r)$.

Supposing that we can find a sufficient statistic $\tau(\bar{\sigma}_r)$, we solve the DP we follows:

1. *Initialization.* As a boundary condition, set $V_{n+1}(\cdot) \equiv 0$.
2. For $r = n, \ldots, 1$ do the following:
   (a) For every feasible value of $t(\bar{\sigma}_r)$, solve the dynamic programming recursion to compute $V_r(\bar{\sigma}_r) = V_r(\tau(\bar{\sigma}_r))$ by replacing $\bar{\sigma}_r$ with $\tau(\bar{\sigma}_r)$ in (EC.1). Such replacement is legal because of our definition that $\tau(\bar{\sigma}_r)$ is a sufficient statistic.
3. Given $V_1(\tau(\bar{\sigma}_1))$ for all possible values of $\tau(\bar{\sigma}_1)$, now solve

$$V_0 = \max_{\sigma_0 \in \mathcal{N} \cup \{0\}} \left\{ \sum_{(\sigma_0, A) \in \mathcal{C}} c_{\sigma_0, A} \quad + \quad V_1\left(\tau(\bar{\sigma}_1)\right) \right\}$$

   Store the optimal solution as σ_0^* and set $\bar{\sigma}_1^* = (\sigma_0^*)$.
4. Run backward induction to determine the optimal solution: for $r = 1, 2, \ldots, n$ do the following:
   (a) $\sigma_r^* = \arg\max_{\sigma_r \in \mathcal{D}_r} V_r(\tau(\bar{\sigma}_r^*))$, where $\mathcal{D}_r$ is the domain of σ_r. If there are multiple optima, pick one solution arbitrarily.
5. The optimal preference list $\boldsymbol{\sigma}^*$ is given by $(\sigma_0^*, \sigma_1^*, \ldots, \sigma_n^*)$.

Note that the above algorithm has two loops that run $n+1$ times and for each r, the computational complexity is bounded above by $O(n\,|\mathcal{C}|\,X_r)$. Therefore, absorbing constant terms into the big-Oh notation, the complexity of the above procedure scales as $O\left(n\,|\mathcal{C}|\sum_{r=0}^{n} X_r\right)$. We now show that the term $\sum_{r=0}^{n} X_r$ scales polynomially in n when the choice, consideration-set combinations belong to $\mathcal{C} \subset \mathcal{C}_d$, for a fixed d. We show this by first exhibiting a sufficient statistic $\tau(\bar{\sigma}_r)$ and then showing that it is polynomially large.

We claim that $\tau(\bar{\sigma}_r)$ defined below is a sufficient statistic:

$$\tau(\bar{\sigma}_r) = \{\sigma_0, \sigma_1, \ldots, \sigma_{r-1}\} \cap \{\sigma_*, \sigma_* + 1, \ldots, \sigma_* + 2d\},$$

where we define $\sigma_* \stackrel{\text{def}}{=} \min\{\sigma_0, \sigma_1, \ldots, \sigma_{r-1}\}$. In order to prove that $\tau(\bar{\sigma}_r)$ is indeed a sufficient statistic, it is sufficient to establish the following three properties:

1. *One-Period Reward Sufficiency*: The collection of sets $\{(\sigma_r, A) \in \mathcal{C}_d \,:\, A \cap \{\sigma_0, \sigma_1, \ldots, \sigma_{r-1}\} = \varnothing\}$ can be determined from $\tau(\bar{\sigma}_r)$.
2. *State Space Sufficiency*: $\tau(\bar{\sigma}_{r+1})$ can be determined from σ_r and $\tau(\bar{\sigma}_r)$.
3. *Action-Set Sufficiency:* The domain of σ_r i.e., $\{0\} \cup (\mathcal{N} \setminus \bar{\sigma}_r)$ can be determined from $\tau(\bar{\sigma}_r)$.

The conditions above ensure that we can choose the optimal action in each stage by keeping track of only the sufficient statistic.

We first show that we can construct the collection $\{(\sigma_r, A) \in \mathcal{C}_d : A \cap \{\sigma_0, \sigma_1, \ldots, \sigma_{r-1}\} = \varnothing\}$ using only $\tau(\bar{\sigma}_r)$. For this, fix a σ_r and consider a subset A such that $(\sigma_r, A) \in \mathcal{C}_d$. Then, it follows from the definition of $\mathcal{C}_d$ that $A = \{a \colon p_a \le b\}$ for some $b \in \mathcal{B}$ and $\boldsymbol{p} \in \mathscr{P}_d$. If π denotes the price ordering corresponding to $\boldsymbol{p}$ so that $p_{\pi^{-1}(0)} \le p_{\pi^{-1}(1)} \le \cdots p_{\pi^{-1}(n)}$ with ties broken arbitrarily, then $A = \{\pi^{-1}(0), \pi^{-1}(1), \ldots, \pi^{-1}(i)\}$ for some integer i. Further, since b is finite, $p_a < \infty$ for all $a \in S_i$. We now have the following claim.

Claim: $\{\pi^{-1}(0), \pi^{-1}(1), \ldots, \pi^{-1}(i)\} \cap \{\sigma_0, \sigma_1, \ldots, \sigma_{r-1}\} = \varnothing$ if and only if $\{\pi^{-1}(0), \pi^{-1}(1), \ldots, \pi^{-1}(i)\} \subseteq \{0, 1, 2, \ldots, \sigma_* + 2d\} \setminus \tau(\bar{\sigma}_r)$, where recall that we are given an indexing of the products so that $|\pi(j) - j| \le d$ for any product j such that $p_j \ne \infty$.

Proof of Claim: For simplicity of notation, let S_i denote the set $\{\pi^{-1}(0), \pi^{-1}(1), \ldots, \pi^{-1}(i)\}$. We prove the claim in two steps: (i) $S_i \cap \{\sigma_0, \sigma_1, \ldots, \sigma_{r-1}\} = \varnothing \implies S_i \subseteq \{0, 1, \ldots, \sigma_* + 2d\} \setminus \tau(\bar{\sigma}_r)$ and (ii) $S_i \subseteq \{0, 1, \ldots, \sigma_* + 2d\} \setminus \tau(\bar{\sigma}_r) \implies S_i \cap \{\sigma_0, \sigma_1, \ldots, \sigma_{r-1}\} = \varnothing$.

Step (i): $S_i \cap \{\sigma_0, \sigma_1, \ldots, \sigma_{r-1}\} = \varnothing \implies S_i \subseteq \{0, 1, \ldots, \sigma_* + 2d\} \setminus \tau(\bar{\sigma}_r)$. We prove the contrapositive of this result. Suppose $S_i \nsubseteq \{0, 1, 2, \ldots, \sigma_* + 2d\} \setminus \tau(\bar{\sigma}_r)$. Then, for some $j < i$, we have that $\pi^{-1}(j) \in \tau(\bar{\sigma}_r) \cup \{\ell \colon \ell > \sigma_* + 2d\}$. If $\pi^{-1}(j) \in \tau(\bar{\sigma}_r)$, then we must have that $\pi^{-1}(j) \in \{\sigma_0, \sigma_1, \ldots, \sigma_{r-1}\}$ because $\tau(\bar{\sigma}_r) \subseteq \{\sigma_0, \sigma_1, \ldots, \sigma_{r-1}\}$. It thus follows that $S_i \cap \{\sigma_0, \sigma_1, \ldots, \sigma_{r-1}\} \ne \varnothing$, proving the result. Now suppose $\pi^{-1}(j) > \sigma_* + 2d$. Since $p_{\pi^{-1}(j)} < \infty$ and $\boldsymbol{p} \in \mathscr{P}_d$, it follows that $|\pi(\pi^{-1}(j)) - \pi^{-1}(j)| \le d$, which implies that $j > -d + \pi^{-1}(j)$. This combined with the fact that $\pi^{-1}(j) > \sigma_* + 2d$ implies that $j > \sigma_* + d > \pi(\sigma_*)$, where the last inequality follows because $|\pi(\sigma^*) - \sigma^*| \le d$. It thus follows that the price rank of σ_* is smaller than j, which in turn implies that $\sigma_* \in S_i$ because S_i contains the products with price ranks from 0 to $i > j$. Further, it follows by our definition that $\sigma_* = \min\{\sigma_0, \sigma_1, \ldots, \sigma_{r-1}\} \in \{\sigma_0, \sigma_1, \ldots, \sigma_{r-1}\}$. Hence, $S_i \cap \{\sigma_0, \sigma_1, \ldots, \sigma_{r-1}\} \supset \{\sigma_*\} \ne \varnothing$. This finishes the proof of the first step.

Step (ii): $S_i \subseteq \{0, 1, \ldots, \sigma_* + 2d\} \setminus \tau(\bar{\sigma}_r) \implies S_i \cap \{\sigma_0, \sigma_1, \ldots, \sigma_{r-1}\} = \varnothing$. It is sufficient to show that for $\ell = 0, 1, \ldots, r-1$, either $\sigma_\ell \in \tau(\bar{\sigma}_r)$ or $\sigma_\ell > \sigma_* + 2d$. It follows from the definition of $\tau(\bar{\sigma}_r)$ that $\sigma_\ell \in \tau(\bar{\sigma}_r)$ if and only if $\sigma_\ell \in \{\sigma_*, \ldots, \sigma_* + 2d\}$. So, suppose $\sigma_\ell \notin \{\sigma_*, \ldots, \sigma_* + 2d\}$. Now since $\sigma_* \stackrel{\text{def}}{=} \min\{\sigma_0, \ldots, \sigma_{r-1}\}$, we have that $\sigma_\ell \ge \sigma_*$. Hence, $\sigma_\ell \notin \{\sigma_*, \ldots, \sigma_* + 2d\}$ must imply that $\sigma_\ell > \sigma_* + 2d$, completing the proof.

The result of the claim now follows. ■

It is now clear from the above claim that we can construct the collection of tuples $\{(\sigma_r, A) \in \mathcal{C}_d : A \cap \{\sigma_0, \sigma_1, \ldots, \sigma_{r-1}\} = \varnothing\}$ can be constructued by including the tuples $(\sigma_r, A) \in \mathcal{C}_d$ for which $A \subset \{0, 1, \ldots, \sigma_* + 2d\} \setminus \tau(\bar{\sigma}_r)$ where $\sigma_* = \min \tau(\bar{\sigma}_r)$. This finishes the proof of *One-Period Reward Sufficiency* property of the sufficient statistic.

We now prove *State Space Sufficiency* property i.e., $\tau(\bar{\sigma}_{r+1})$ can be determined from σ_r and $\tau(\bar{\sigma}_r)$. There are two cases to consider: (1) $\sigma_r > \sigma_*$ and (2) $\sigma_r < \sigma_*$ where $\sigma_* = \min\{\sigma_0, \ldots, \sigma_{r-1}\}$. In the case $\sigma_r > \sigma_*$, we have that σ_* remains the minimum even after adding σ_r i.e., we have that $\sigma_* = \min\{\sigma_0, \sigma_1, \ldots, \sigma_{r-1}, \sigma_r\}$. It now immediately follows that

$$\tau(\bar{\sigma}_{r+1}) = \{\sigma_0, \sigma_1, \ldots, \sigma_{r-1}, \sigma_r\} \cap \{\sigma_*, \sigma_* + 1, \ldots, \sigma_* + 2d\}$$

$$= (\{\sigma_r\} \cup (\{\sigma_0, \sigma_1, \ldots, \sigma_{r-1}\}) \cap \{\sigma_*, \sigma_* + 1, \ldots, \sigma_* + 2d\})$$
$$= (\{\sigma_r\} \cap \{\sigma_*, \sigma_* + 1, \ldots, \sigma_* + 2d\}) \cup \tau(\sigma_r).$$

Now suppose $\sigma_r < \sigma_*$. This implies that $\sigma_r = \min\{\sigma_0, \sigma_1, \ldots, \sigma_{r-1}, \sigma_r\}$. Thus, we can write

$$\begin{aligned}
\tau(\bar{\sigma}_{r+1}) &= \{\sigma_1, \ldots, \sigma_{r-1}, \sigma_r\} \cap \{\sigma_r, \sigma_r + 1, \ldots, \sigma_r + 2d\} \\
&= \{\sigma_r\} \cup (\{\sigma_1, \ldots, \sigma_{r-1}\} \cap \{\sigma_r + 1, \ldots, \sigma_r + 2d\}) \\
&= \{\sigma_r\} \cup (\{\sigma_1, \ldots, \sigma_{r-1}\} \cap \{\sigma_*, \ldots, \sigma_r + 2d\}) \\
&= \{\sigma_r\} \cup (\tau(\bar{\sigma}_r) \setminus \{\sigma_r + 2d + 1, \ldots, \sigma_* + 2d\}),
\end{aligned}$$

where the third equality follows from the fact that $\sigma_\ell \notin \{\sigma_r + 1, \ldots, \sigma_* - 1\}$ for all $1 \le \ell \le r - 1$ because $\sigma_* = \min\{\sigma_0, \sigma_1, \ldots, \sigma_{r-1}\}$. It thus follows that $\tau(\bar{\sigma}_{r+1})$ is completely determined by σ_r and $\tau(\bar{\sigma}_{i-1})$. This establishes *State Space Sufficiency*.

We now focus on *Action-Set Sufficiency*. For that, we consider two cases: (i) $\max \tau(\bar{\sigma}_r) < n - r + |\tau(\bar{\sigma}_r)|$ and (ii) $\max \tau(\bar{\sigma}_r) = n - r + |\tau(\bar{\sigma}_r)|$. We claim that in case (i), σ_r can take any value in $\{0\} \cup (\mathcal{N} \setminus \tau(\bar{\sigma}_r))$ and in case (ii), σ_r can take any value in $\{0, 1, \ldots, \max \tau(\bar{\sigma}_r)\} \setminus \tau(\bar{\sigma}_r)$.

To see the above, first consider case (i), so that $\max \tau(\bar{\sigma}_r) < n - r + |\tau(\bar{\sigma}_r)|$. Since $\tau(\bar{\sigma}_r) \subseteq \{\sigma_0, \sigma_1, \ldots, \sigma_{r-1}\}$ and σ_r must not take any values in $\{\sigma_0, \sigma_1, \ldots, \sigma_{r-1}\}$, it is clear that $\sigma_r \notin \tau(\bar{\sigma}_r)$. Now consider any element $x \in (\mathcal{N} \setminus \tau(\bar{\sigma}_r)) \cup \{0\}$. If $x < \max \tau(\bar{\sigma}_r)$, it is clear that $x \notin \{\sigma_0, \sigma_1, \ldots, \sigma_{r-1}\}$ because any $\sigma_\ell \le \max \tau(\bar{\sigma}_r)$ must belong to the set $\tau(\bar{\sigma}_r)$. Now suppose $x > \max \tau(\bar{\sigma}_r)$. If $\max \tau(\bar{\sigma}_r) = \sigma_{r-1}$, then it immediately follows that $x \notin \{\sigma_0, \sigma_1, \ldots, \sigma_{r-1}\}$.

So, we suppose that $\max \tau(\bar{\sigma}_r) = \sigma_* + 2d$. Now consider any $\sigma_r > \sigma_* + 2d$. We claim that the objective value of the optimization problem in the DP recursion is given by

$$\sum_{(\sigma_r, A) \in \mathcal{C}_d \colon A \cap \bar{\sigma}_r = \emptyset} c_{\sigma_r, A} + V_{r+1}(\tau(\bar{\sigma}_{r+1})) = V_{r+1}(\tau(\bar{\sigma}_r)). \tag{EC.2}$$

To see this, first note that it follows from the arguments in the proof of *State Space Sufficiency* that $\tau(\bar{\sigma}_{r+1}) = \tau(\bar{\sigma}_r)$. Therefore, we have $V_{r+1}(\tau(\bar{\sigma}_{r+1})) = V_{r+1}(\tau(\bar{\sigma}_r))$. Now consider a choice, consideration-set tuple $(\sigma_r, A) \in \mathcal{C}_d$. Since $\sigma_r \in A$ and $\sigma_r > \sigma_* + 2d$, it must be that $A \nsubseteq \{0, 1, \ldots, \sigma_* + 2d\} \setminus \tau(\bar{\sigma}_r)$. Therefore, it follows from the claim in the proof of *One-Period Reward Sufficiency* that $A \cap \{\sigma_0, \sigma_1, \ldots, \sigma_{r-1}\} \neq \varnothing$. Thus, the set $\{(\sigma_r, A) \in \mathcal{C}_d \colon A \cap \bar{\sigma}_r = \emptyset\}$ is an empty set. This establishes the equality (EC.2).

It now follows from (EC.2) that the objective value is the same for all $\sigma_r > \sigma_* + 2d$. Hence, if the optimal solution in the DP recursion is such that $V_r(\tau(\bar{\sigma}_r)) = V_{r+1}(\tau(\bar{\sigma}_r))$, then it follows that any $\sigma_r > \sigma_* + 2d$ is an optimal solution. Since there are multiple optimal solution, we can pick one solution arbitrarily. We argue that we can always pick a solution σ_r such that $\sigma_r \notin \{\sigma_0, \sigma_1, \ldots, \sigma_{r-1}\}$. To see this, consider the set $\{\sigma_* + 2d + 1, \ldots, n\}$. This set contains $n - \sigma_* - 2d$ elements. Of this number, $r - |\tau(\bar{\sigma}_r)|$ elements must belong to the set $\{\sigma_0, \sigma_1, \ldots, \sigma_{r-1}\}$. Hence, we can always find a $\sigma_r > \sigma_* + 2d$ such that $\sigma_r \notin \{\sigma_0, \sigma_1, \ldots, \sigma_{r-1}\}$ if and only if $n - \sigma_* - 2d > r - |\tau(\bar{\sigma}_r)|$, or equivalently, $\sigma_* + 2d < n - r + |\tau(\bar{\sigma}_r)|$. This condition is indeed true because of our assumption that $\max \tau(\bar{\sigma}_r) = \sigma_* + 2d$. This establishes case (i).

Now conider case (ii), so that $\max\tau(\bar{\sigma}_r) = n - r + |\tau(\bar{\sigma}_r)|$. In this case, we claim that σ_r can take values in $\{0,1,\ldots,\max\tau(\bar{\sigma}_r)\} \setminus \tau(\bar{\sigma}_r)$. First note that it follows from our definitions that $(\{0,1,\ldots,\max\tau(\bar{\sigma}_r)\} \setminus \tau(\bar{\sigma}_r)) \cap \{\sigma_0,\sigma_1,\ldots,\sigma_{r-1}\} = \varnothing$. Hence, $\sigma_r \in \{0,1,\ldots,\max\tau(\bar{\sigma}_r)\} \setminus \tau(\bar{\sigma}_r)$ ensures that $\sigma_r \notin \{\sigma_0,\sigma_1,\ldots,\sigma_{r-1}\}$. Further, we claim that $\{\max\tau(\bar{\sigma}_r),\ldots,n\} \subseteq \{\sigma_0,\ldots,\sigma_{r-1}\}$. If this claim is true, then it is clear that σ_r cannot be outside the set $\{0,1,\ldots,\max\tau(\bar{\sigma}_r)\}$, establishing this case. To see why the claim is true, note that the cardinality of the set $\{\max\tau(\bar{\sigma}_r),\ldots,n\}$ is $n-\max\tau(\bar{\sigma}_r)+1$, which is equal to $r-|\tau(\bar{\sigma}_r)|$ because $max\tau(\bar{\sigma}_r) = n-r+|\tau(\bar{\sigma}_r)|$. Now consider the set $\{\sigma_0,\sigma_1,\ldots,\sigma_{r-1}\} \setminus \tau(\bar{\sigma}_r)$, which has cardinality $r-|\tau(\bar{\sigma}_r)|$. Thus, both the sets $\{\sigma_0,\sigma_1,\ldots,\sigma_{r-1}\} \setminus \tau(\bar{\sigma}_r)$ and $\tau(\bar{\sigma}_r) \subseteq \{\max\tau(\bar{\sigma}_r),\ldots,n\}$ have the same cardinality. In addition, it follows from our definitions that $\{\sigma_0,\sigma_1,\ldots,\sigma_{r-1}\} \setminus \tau(\bar{\sigma}_r) \subseteq \{\max\tau(\bar{\sigma}_r),\ldots,n\}$. As a result, we can conclude that $\{\max\tau(\bar{\sigma}_r),\ldots,n\} = \{\sigma_0,\sigma_1,\ldots,\sigma_{r-1}\} \setminus \tau(\bar{\sigma}_r)$.

We have thus established the *Action-Set Sufficiency* property.

We are now left with determining the number of distinct values that $\tau(\bar{\sigma}_r)$ can take. Since $\tau(\bar{\sigma}_r) \subseteq \{\sigma_*,\ldots,\sigma_*+2d\}$, it can take at most 2^{2d+1} distinct values for a given σ_*. Further σ_* can take at most n values. Together, we can conclude that $\tau(\bar{\sigma}_r)$ can take at most $n2^{2d+1}$ distinct values. Following our notation above, we have shown that $X_r \le n2^{2d+1}$ for any r. We must thus have that $\sum_{r=0}^{n} X_r = O(n^2 4^d)$. So, the total complexity is $O(x\,|\mathcal{C}|\sum_{r=0}^{n} X_r) = O\left(n^3\,|\mathcal{C}|\,4^d\right)$.

The result of the proposition now follows.

Appendix C: Proofs for Section 5

In this section, we will prove the PTAS for the joint assortment and price optimization problem given in Theorem 5.1. We first show that restricting the prices to a discrete domain Dom_α results in a minor loss in performance (see Lemma C.1). Then, we show that the solution to the relaxed revenue function $R^{\alpha,k}(\cdot)$ gives the desired performance guarantee for the original optimization problem (see Lemma C.2). Then, in Section C.1, we show that optimizing the relaxed revenue function can be done using dynamic programming, and establish the running time of the DP, proving Proposition 5.1. Then, in Section C.2, we combine all of the results together and prove Theorem 5.1. Throughout this section, we assume that the products are indexed by $1,\ldots,n$, and the reference rank of product i is i; that is $\boldsymbol{\tau}$ is the identity ordering. Also, recall that we assume that $\max_{b\in\mathcal{B}} b < 1$, and thus, pricing a product at 1 effectively removes it from the offer set. Therefore, the d-sorted family of prices is given by

$$\mathscr{P}_d = \left\{ \boldsymbol{p} \in [0,1]^n \;:\; \max_{i\,:\,p_i<1} \left|\pi_{\boldsymbol{p}}^{-1}(i) - i\right| \le d \right\}$$

where $\pi_{\boldsymbol{p}}$ represents the price ordering under $\boldsymbol{p}$, with $p_{\pi_{\boldsymbol{p}}(1)} \le p_{\pi_{\boldsymbol{p}}(2)} \le \cdots \le p_{\pi_{\boldsymbol{p}}(n)}$, and for any i, $\pi_{\boldsymbol{p}}(i)$ denotes the product at rank i under $\boldsymbol{p}$. Note that $\pi_{\boldsymbol{p}}^{-1}(i)$ denote the price rank of product i.

The following lemma shows that by restricting our search to a discrete set $\mathscr{P}_{d,\alpha}$, the maximum revenue decreases by at most a factor of α.

Lemma C.1 **(Rounding)**. *For any $\alpha \in (0,1)$ and $\boldsymbol{p} \in \mathscr{P}_d$, there exists $\hat{\boldsymbol{p}} \in \mathscr{P}_{d,\alpha}$ such that $R(\hat{\boldsymbol{p}}) \ge \alpha R(\boldsymbol{p})$. Consequently, $\max_{\boldsymbol{p}\in\mathscr{P}_{d,\alpha}} R(\boldsymbol{p}) \ge \alpha Z^*$.*

Proof: For all i, let $\hat{p}_i = \max\{x \in \mathsf{Dom}_\alpha : x \le p_i\}$. It suffices to show that for any $(b, \boldsymbol{\sigma})$,

$$\sum_{i=1}^n \hat{p}_i \mathbb{1}[\boldsymbol{\sigma}, i, \{\ell : \hat{p}_\ell \le b\}] \;\ge\; \alpha \sum_{i=1}^n p_i \mathbb{1}[\boldsymbol{\sigma}, i, \{\ell : p_\ell \le b\}]$$

because summing over all $(b, \boldsymbol{\sigma})$ implies that that $R(\hat{\boldsymbol{p}}) \ge \alpha R(\boldsymbol{p})$. Let $X = \{\ell : p_\ell \le b\}$ and $\hat{X} = \{\ell : \hat{p}_\ell \le b\}$. Since $\hat{p}_i \le p_i$ for all i, we have that $X \subseteq \hat{X}$. Let $c = \arg\min_{\ell \in X} \sigma^{-1}(\ell)$ denote the choice under $\boldsymbol{p}$, and let $s = \arg\min_{\ell \in \hat{X}} \sigma^{-1}(\ell)$ denote the choice under $\hat{\boldsymbol{p}}$. If $c = s$, then the above inequality is equivalent to $\hat{p}_c \ge \alpha p_c$, which is true by our construction. So, consider the case where $s \ne c$. Since $X \subseteq \hat{X}$, it must be the case that $s \in \hat{X} \setminus X$, the choice s must have the property that $p_s > b \ge \hat{p}_s$; otherwise, s would have be chosen under $\boldsymbol{p}$. Moreover, by definition of c, $b \ge p_c$. Thus, $\hat{p}_s \ge \alpha p_s > \alpha b \ge \alpha p_c$, which is the desired result. ∎

The next proposition relates the solution of the relaxed revenue function to the solution of the original revenue function.

Proposition C.1 **(Approximation Guarantee)**. *For any $\alpha \in (0,1)$ and $k \in \mathbb{Z}_{++}$, if $\boldsymbol{p}^{\alpha,k}$ is an optimal solution to the optimization problem associated with the relaxed revenue function $\max_{\boldsymbol{p} \in \mathscr{P}_{d,\alpha}} R^{\alpha,k}(\boldsymbol{p})$, then*

$$Z^* \ge R(\boldsymbol{p}^{\alpha,k}) \ge \alpha(1 - \alpha^{k+1}) Z^* \; .$$

In order to prove Proposition C.1, we need the next lemma, which shows that $R^{\alpha,k}$ is close to R.

Lemma C.2 **(Relaxed Revenue Function)**. *For all $\alpha \in (0,1)$, $k \in \mathbb{Z}_{++}$, and $\boldsymbol{p} \in \mathscr{P}_{d,\alpha}$,*

$$\left(1 - \alpha^{k+1}\right) R^{\alpha,k}(\boldsymbol{p}) \le R(\boldsymbol{p}) \le R^{\alpha,k}(\boldsymbol{p}) \quad .$$

Proof: Note that $R(\boldsymbol{p}) \le R^{\alpha,k}(\boldsymbol{p})$ by our construction. To prove the remaining inequality, it suffices to show the result for a single customer type. For any $(b, \boldsymbol{\sigma})$, we will show that

$$\sum_{i=1}^n p_i \mathbb{1}\left[\boldsymbol{\sigma}, i, \left\{\ell : p_\ell \le b \text{ and } p_\ell \le \frac{p_i}{\alpha^k}\right\}\right] \le \frac{1}{1 - \alpha^{k+1}} \sum_{i=1}^n p_i \mathbb{1}[\boldsymbol{\sigma}, i, \{\ell : p_\ell \le b\}]$$

Let $c = \arg\min\{\sigma^{-1}(\ell) \;:\; p_\ell \le b\}$ denote the choice of the customer under the original revenue function.

Claim: For any i such that $p_i \ge p_c$, $\mathbb{1}\left[\boldsymbol{\sigma}, i, \left\{\ell : p_\ell \le b \text{ and } p_\ell \le \frac{p_i}{\alpha^k}\right\}\right] = 0$

If $c = 0$, then the claim is trivially true. So, suppose that $c \ne 0$. For any i such that $p_i \ge p_c$, note that $p_c < p_i/\alpha^k$. So, $c \in \left\{\ell : p_\ell \le b \text{ and } p_\ell \le \frac{p_i}{\alpha^k}\right\}$ and c is the most preferred product within the set $\{\ell : p_\ell \le b\}$. Therefore, product i will never be chosen.

$$\begin{aligned}
\sum_{i=1}^n p_i \mathbb{1}\left[\boldsymbol{\sigma}, i, \left\{\ell : p_\ell \le b \text{ and } p_\ell \le \frac{p_i}{\alpha^k}\right\}\right] &= \sum_{i : p_i \le p_c} p_i \mathbb{1}\left[\boldsymbol{\sigma}, i, \left\{\ell : p_\ell \le b \text{ and } p_\ell < \frac{p_i}{\alpha^k}\right\}\right] \\
&\le p_c + p_c \alpha^{k+1} + p_c \alpha^{2(k+1)} + p_c \alpha^{3(k+1)} + \cdots = \frac{p_c}{1 - \alpha^{k+1}} \; ,
\end{aligned}$$

where the last inequality follows from the fact that the the next most expensive product that can be purchased under the *relaxed* revenue function is the one with price $p_c \alpha^{k+1}$, and the one after that has a price at most $p_c \alpha^{2(k+1)}$, etc. This is the desired result. ∎

Proof of Proposition C.1: Then, it follows from Lemma C.2 and the above bound that

$$\begin{aligned}
R(\boldsymbol{p}^{\alpha,k}) &\ge (1 - \alpha^{k+1})\; R^{\alpha,k}(\boldsymbol{p}^{\alpha,k}) = (1 - \alpha^{k+1}) \max_{\boldsymbol{p} \in \mathscr{P}_{d,\alpha}} R^{\alpha,k}(\boldsymbol{p}) \\
&\ge (1 - \alpha^{k+1}) \max_{\boldsymbol{p} \in \mathscr{P}_{d,\alpha}} R(\boldsymbol{p}) \ge \alpha(1 - \alpha^{k+1}) \max_{\boldsymbol{p} \in \mathscr{P}_d} R(\boldsymbol{p}) = \alpha(1 - \alpha^{k+1}) Z^*
\end{aligned}$$

∎

C.1. Proof of Proposition 5.1

In this section, we show how to optimize $\max_{\boldsymbol{p}\in\mathscr{P}_{d,\alpha}} R^{\alpha,k}(\boldsymbol{p})$ efficiently using dynamic programming formulation. Note that we are restricting our attention to the discrete domain $\mathscr{P}_{d,\alpha}$. Let α^H denote the largest price that is less than or equal to the smallest budget; that is, $\alpha^H = \max\{\alpha^s : \alpha^s \leq \min_{b\in\mathcal{B}} b\}$, or equivalently, $H = \min\{s : \alpha^s \leq \min_{b\in\mathcal{B}} b\}$. Note that, by our construction, it is never optimal to consider prices less than α^H. Thus,

$$\max_{\boldsymbol{p}\in\mathscr{P}_{d,\alpha}} R^{\alpha,k}(\boldsymbol{p}) = \max_{\boldsymbol{p}\in\mathscr{P}_{d,\alpha}: p_i \geq \alpha^H \ \forall i} R^{\alpha,k}(\boldsymbol{p})$$

The proof of Proposition 5.1 makes use of a series of lemmas. As the first step, we will consider an alternative representation of prices.

An Equivalent Representation of Prices: Since we are working with discrete prices of the form α^s with $0 \leq s \leq H$, each price vector $\boldsymbol{p}$ can be represented as a vector $(A_H, A_{H-1}, \ldots, A_1, A_0)$, where for all s, $A_s = \{i : p_i = \alpha^s\}$ is the set of products whose prices are equal to α^s. Throughout this section, we will use this representation of prices. Note that it is possible that $A_s = \varnothing$ for some s. Also, since the maximum budget is less than one, $A_0 = \{i : p_i = \alpha^0 = 1\}$ effectively correspond to the set of products that we will *not* offer.

An Equivalent Optimization Problem: We will now rephrase the optimization problem associated with $R^{\alpha,k}$ as follows: For any $(A_H, A_{H-1}, \ldots, A_1, A_0)$ and $0 \leq s \leq H$, let $W_s^{\alpha,k}(A_H, A_{H-1}, \ldots, A_1, A_0)$ denote the total revenue under the relaxed revenue function that is collected from the products in $A_s \cup A_{s-1} \cup \cdots A_1 \cup A_0$; that is,

$$\begin{aligned}
&W_s^{\alpha,k}(A_H, A_{H-1}, \ldots, A_1, A_0) \\
&\stackrel{\text{def}}{=} \sum_{(b,\boldsymbol{\sigma})} w(b,\boldsymbol{\sigma}) \sum_{\ell=1}^{s} \alpha^\ell \sum_{i\in A_\ell} \mathbb{1}\left[\boldsymbol{\sigma}, i, \left\{q \in \{0\} \cup A_H \cup A_{H-1} \cup \cdots \cup A_1 \cup A_0 : p_q \leq b \text{ and } p_q \leq \frac{p_i}{\alpha^k}\right\}\right] \\
&= \sum_{(b,\boldsymbol{\sigma})} w(b,\boldsymbol{\sigma}) \sum_{\ell=1}^{s} \alpha^\ell \sum_{i\in A_\ell} \mathbb{1}[\boldsymbol{\sigma}, i, \{q \in \{0\} \cup A_H \cup A_{H-1} \cup \cdots \cup A_{\ell-k} : p_q \leq b\}]
\end{aligned}$$

Note that since $b < 1$ for all $b \in \mathcal{B}$, products in A_0 are never selected. For $s = 1, \ldots, H$, let

$$Y_s^* \stackrel{\text{def}}{=} \max_{(A_H, A_{H-1}, \ldots, A_1, A_0)\in\mathscr{P}_{d,\alpha}} W_s^{\alpha,k}(A_H, A_{H-1}, \ldots, A_1, A_0)$$

We note that the objective $W_s^{\alpha,k}$ is slightly different from $R^{\alpha,k}$ because we only compute revenue from products whose prices are α^s or higher; that is, $\alpha^s, \alpha^{s-1}, \ldots, \alpha$. The following lemma shows that Y_H^* is our desired target.

Lemma C.3 **(Equivalent Optimization)**. $Y_H^* = \max_{\boldsymbol{p}\in\mathscr{P}_{d,\alpha}} R^{\alpha,k}(\boldsymbol{p})$.

Proof: Since it is never optimal to price below α^H, we have that

$$\max_{\boldsymbol{p}\in\mathscr{P}_{d,\alpha}} R^{\alpha,k}(\boldsymbol{p}) = \max_{\boldsymbol{p}\in\mathscr{P}_{d,\alpha}: p_i \geq \alpha^H \ \forall i} R^{\alpha,k}(\boldsymbol{p}) = \max_{(A_H, A_{H-1}, \ldots, A_1, A_0)\in\mathscr{P}_{d,\alpha}} W_H^{\alpha,k}(A_H, A_{H-1}, \ldots, A_1, A_0) = Y_H^*,$$

which is the desired result. ■

Based on Lemma C.3, we will develop dynamic programming methods for computing $Y_0^*, Y_1^*, \ldots, Y_H^*$. Define the following the value functions. For $s = 0, 1, \ldots, H$, let

$$\begin{aligned}&J_s^*(A_H, A_{H-1}, \ldots, A_s, A_{s-1}, \ldots, A_{s-k+2}, A_{s-k+1})\\ &\stackrel{\text{def}}{=} \max_{(A_{s-k},\ldots,A_0)} \left\{ W_s^{\alpha,k}(A_H, A_{H-1}, \ldots, A_s, \ldots, A_{s-k+2}, A_{s-k+1}, A_{s-k}, \ldots, A_0) \;\middle|\; (A_H, A_{H-1}, \ldots, A_{s-k+1}, A_{s-k}, \ldots, A_0) \in \mathscr{P}_{d,\alpha} \right\}\end{aligned}$$

Note that

$$\max_{(A_H,\ldots,A_{s-k+2},A_{s-k+1})} J_s^*(A_H, A_{H-1}, \ldots, A_{s-k+2}, A_{s-k+1}) = \max_{(A_H,A_{H-1},\ldots,A_1,A_0)\in\mathscr{P}_{d,\alpha}} W_s^{\alpha,k}(A_H, A_{H-1}, \ldots, A_1, A_0) = Y_s^*$$

The following lemma provides a dynamic programming recursion for computing the value function.

Lemma C.4 **(Dynamic Programming Equation)**. *For any s,*

$$\begin{aligned}&J_s^*(A_H, A_{H-1}, \ldots, A_s, A_{s-1}, \ldots, A_{s-k+2}, A_{s-k+1})\\ &= \max_{A_{s-k}\in\mathcal{D}_s(A_H,A_{H-1},\ldots,A_{s-k+2},A_{s-k+1})} \Big\{ \alpha^s G_s(A_H, \ldots, A_{s-1}, A_s, \ldots, A_{s-k+1}, A_{s-k})\\ &\qquad\qquad + J_{s-1}^*(A_H, A_{H-1}, \ldots, A_s, A_{s-1}, A_{s-2}, \ldots, A_{s-k+1}, A_{s-k}) \Big\},\end{aligned}$$

where

$$G_s(A_H, \ldots, A_{s-1}, A_s, \ldots, A_{s-k+1}, A_{s-k}) = \sum_{(b,\boldsymbol{\sigma})} w(b,\boldsymbol{\sigma}) \sum_{i\in A_s} \mathbb{1}\left[\boldsymbol{\sigma}, i, \{q \in \{0\}\cup A_H \cup A_{H-1} \cup\cdots\cup A_{s-k} : p_q \le b\}\right],$$

*and the set $\mathcal{D}_s(A_H, A_{H-1}, \ldots, A_{s-k+2}, A_{s-k+1})$ denotes the collection of subset of products that can be priced at α^{s-k} and still satisfy the d-*sorted *constraint; that is,*

$$\begin{aligned}&\mathcal{D}_s(A_H, A_{H-1}, \ldots, A_{s-k+2}, A_{s-k+1})\\ &\stackrel{\text{def}}{=} \left\{ X \subseteq \left(\cup_{\ell=H}^{s-k+1} A_\ell\right)^c : \max_{\ell\in X} \left|\boldsymbol{\pi}^{-1}(\ell) - \ell\right| \le d \right\}\\ &= \left\{ X \;:\; X = \{i_1, \ldots, i_q\} \subseteq \left(\cup_{\ell=H}^{s-k+1} A_\ell\right)^c \text{ for some } i_1 < i_2 < \cdots < i_q, \text{ and } \max_{u=1,\ldots,q} \left| \left(\sum_{\ell=H}^{s-k+1} |A_\ell| + u \right) - i_u \right| \le d \right\},\end{aligned}$$

where the last equality follows from the fact that the rank of the first product in A_{s-k} is equal to $1 + \sum_{\ell=H}^{s-k+1} |A_\ell|$.

Proof: By definition,

$$\begin{aligned}&W_s^{\alpha,k}(A_H, A_{H-1}, \ldots, A_1, A_0) - W_{s-1}^{\alpha,k}(A_H, A_{H-1}, \ldots, A_1, A_0)\\ &= \alpha^s \sum_{(b,\boldsymbol{\sigma})} w(b,\boldsymbol{\sigma}) \sum_{i\in A_s} \mathbb{1}\left[\boldsymbol{\sigma}, i, \{q \in \{0\}\cup A_H \cup A_{H-1} \cup\cdots\cup A_{s-k} : p_q \le b\}\right] \;=\; \alpha^s G_s(A_H, \ldots, A_{s-1}, A_s, \ldots, A_{s-k+1}, A_{s-k})\end{aligned}$$

Therefore, it follows from the definition of J_s^* that

$$\begin{aligned}&J_s^*(A_H, A_{H-1}, \ldots, A_s, A_{s-1}, \ldots, A_{s-k+2}, A_{s-k+1})\\ &= \max_{(A_{s-k},\ldots,A_0)} \Big\{ \alpha^s G_s(A_H, \ldots, A_{s-1}, A_s, \ldots, A_{s-k+1}, A_{s-k}) + W_{s-1}^{\alpha,k}(A_H, A_{H-1}, \ldots, A_1, A_0) \;\Big|\\ &\qquad\qquad (A_H, A_{H-1}, \ldots, A_{s-k+2}, A_{s-k+1}, A_{s-k}, \ldots, A_0) \in \mathscr{P}_{d,\alpha} \Big\}\end{aligned}$$

$$= \max_{A_{s-k}} \left\{ \alpha^s G_s(A_H, \ldots, A_{s-1}, A_s, \ldots, A_{s-k+1}, A_{s-k}) + \max_{(A_{s-k-1}, \ldots, A_0)} W^{\alpha,k}_{s-1}(A_H, A_{H-1}, \ldots, A_{s-k+1}, A_{s-k}, A_{s-k-1} \ldots, A_1, A_0) \ \middle| \right.$$
$$\left. (A_H, A_{H-1}, \ldots, A_{s-k+1}, A_{s-k}, A_{s-k-1}, \ldots, A_0) \in \mathscr{P}_{d,\alpha} \right\}$$
$$= \max_{A_{s-k} \in \mathcal{D}_s(A_H, A_{H-1}, \ldots, A_{s-k+2}, A_{s-k+1})} \left\{ \alpha^s G_s(A_H, \ldots, A_{s-1}, A_s, \ldots, A_{s-k+1}, A_{s-k}) + J^*_{s-1}(A_H, A_{H-1}, \ldots, A_{s-k+1}, A_{s-k}) \right\},$$

which is the desired result. ■

Sufficient Statistics for the DP in Lemma C.4: We will now show that the DP equation in Lemma C.4 has a tractable sufficient statistics. To facilitate our notation, for any $H \geq j \geq i \geq 0$, let $A_{[j,i]} = (A_j, A_{j-1}, \ldots, A_i)$. As a convention, if $H \geq s \geq 0 > i$, then $A_{[s,i]} = A_{[s,0]}$, and the index for union and summation will start at H and decrease toward zero. The DP equation can be written as

$$J^*_s(A_{[H:s-k+1]}) = \max_{A_{s-k} \in \mathcal{D}_s\left(A_{[H:s-k+1]}\right)} \left\{ \alpha^s G_s(A_{[H:s-k]}) + J^*_{s-1}(A_{[H:s-k]}) \right\}, \tag{EC.3}$$

For any $A_{[H:s-k+1]}$, let

$$\tau(A_{[H:s-k+1]}) = \left(\sum_{\ell=H}^{s-k+1} |A_\ell|,\ L,\ \left(\dot{\bigcup_{\ell=H}^{s-k+1}} A_\ell \right)^c \cap \{L-2d, L-2d+1, \ldots, L\},\ A_{[s:s-k+1]},\ \dot{\bigcup_{\ell=H}^{s}} A_\ell \right),$$

where $L = \max\{\ell : \ell \in \dot\bigcup_{\ell=H}^{s-k+1} A_\ell\}$. We have the following lemma.

LEMMA C.5. $\tau(A_{[H:s-k+1]})$ *as defined above are sufficient statistics for the DP equation* (EC.3).

Proof: In order to prove that $\tau(A_{[H:s-k+1]})$ is indeed a sufficient statistic, it is sufficient to establish the following three properties:

1. *One-Period Reward Sufficiency*: The function $G_s(A_{[H:s-k]})$ can be determined from $\tau(A_{[H:s-k+1]})$.
2. *Action-Set Sufficiency:* The domain $\mathcal{D}_s\left(A_{[H:s-k+1]}\right)$ can be determined from $\tau(A_{[H:s-k+1]})$.
3. *State Space Sufficiency*: $\tau(A_{[H:s-k]})$ can be determined from A_{s-k} and $\tau(A_{[H:s-k+1]})$.

The conditions above ensure that we can choose the optimal action in each stage by keeping track of only the sufficient statistic.

<u>One-Period Reward Sufficiency:</u> Note that

$$G_s(A_H, \ldots, A_{s-1}, A_s, \ldots, A_{s-k+1}, A_{s-k}) = \sum_{(b,\boldsymbol{\sigma})} w(b,\boldsymbol{\sigma}) \sum_{i \in A_s} \mathbb{1}[\boldsymbol{\sigma}, i, \{q \in \{0\} \cup A_H \cup A_{H-1} \cup \cdots \cup A_{s-k} : p_q \leq b\}],$$

We will show that for each customer type $(b,\boldsymbol{\sigma})$ and for each $i \in A_s$, we can determine the value of

$$\mathbb{1}[\boldsymbol{\sigma}, i, \{q \in \{0\} \cup A_H \cup A_{H-1} \cup \cdots \cup A_{s-k} : p_q \leq b\}]$$

using the sufficient statistics. Note that

$$\begin{aligned} &\mathbb{1}[\boldsymbol{\sigma}, i, \{q \in \{0\} \cup A_H \cup A_{H-1} \cup \cdots \cup A_{s-k} : p_q \leq b\}] = 1 \\ \Leftrightarrow\quad & \alpha^s \leq b \quad \text{and} \quad \sigma^{-1}(i) \ < \ \min\left\{ \sigma^{-1}(q) : q \in \dot{\bigcup_{\ell=H}^{s}} A_\ell,\ q \neq i \right\} \\ & \qquad \text{and} \quad \sigma^{-1}(i) \ < \ \min_{\ell : s-1 \leq \ell \leq s-k,\ \alpha^\ell \leq b} \min\left\{ \sigma^{-1}(q) : q \in A_\ell \right\} \end{aligned}$$

Note that we keep track of $\dot\bigcup_{\ell=H}^{s} A_\ell$ and $A_{s-1},\ldots,A_{s-k}$ as part of our sufficient statistics. Thus, this establishes the one-period reward sufficiency.

Action Set Sufficiency: The domain $\mathcal{D}_s(A_{[H:s-k+1]})$ is defined as:

$$\begin{aligned}
&\mathcal{D}_s(A_{[H:s-k+1]})\\
&= \left\{X \subseteq \left(\cup_{\ell=H}^{s-k+1} A_\ell\right)^c : \max_{\ell\in X}\left|\boldsymbol{\pi}^{-1}(\ell)-\ell\right| \le d\right\}\\
&= \left\{X \;:\; X=\{i_1,\ldots,i_q\} \subseteq \left(\cup_{\ell=H}^{s-k+1} A_\ell\right)^c \text{ for some } \; i_1<i_2<\cdots<i_q,\, \max_{u=1,\ldots,q}\left|\left(\sum_{\ell=H}^{s-k+1}|A_\ell|+u\right)-i_u\right| \le d\right\}
\end{aligned}$$

where the last equality follows from the fact that the rank of the first product in A_{s-k} is equal to $1+\sum_{\ell=H}^{s-k+1}|A_\ell|$. Note that we keep track of $\sum_{\ell=H}^{s-k+1}|A_\ell|$ as part of our sufficient statistics.

As a first step, we claim that

$$\mathcal{D}_s(A_{[H:s-k+1]}) = \left\{X \subseteq \left(\cup_{\ell=H}^{s-k+1} A_\ell\right)^c \cap \{q : q \ge L-2d\} \;:\; \max_{\ell\in X}\left|\boldsymbol{\pi}^{-1}(\ell)-\ell\right| \le d\right\}$$

To see this, note that the right-hand-side (RHS) is clearly a subset of $\mathcal{D}_s(A_{[H:s-k+1]})$. So, to establish the claim, it suffices to show that any product i such that $i<L-2d$ cannot be a part of any set in $\mathcal{D}_s(A_{[H:s-k+1]})$. To see this, let $V=\sum_{\ell=H}^{s-k+1}|A_\ell|$. Recall that $L=\max\{\ell : \ell\in \dot\bigcup_{\ell=H}^{s-k+1} A_\ell\}$. By the d-sorted definition, it must be the case that the rank of L is least $L-d$. Thus, we have that $L-d\le V$. Therefore, $i<L-2d\le L-d\le V$, which implies that

$$V-i \ge L-d-i \;>\; L-d-(L-2d)=d\;,$$

which implies that i can never be an element of any set in $\mathcal{D}_s(A_{[H:s-k+1]})$, which establishes the claim.

To complete the proof of the action set sufficiency, note that by definition, $\cup_{\ell=H}^{s-k+1} A_\ell \subseteq \{1,\ldots,L\}$, which implies that $\left(\cup_{\ell=H}^{s-k+1} A_\ell\right)^c \supseteq \{L+1,L+2,\ldots,n\}$. Therefore,

$$\begin{aligned}
\left(\cup_{\ell=H}^{s-k+1} A_\ell\right)^c \cap \{q : q\ge L-2d\} \;&=\; \left(\cup_{\ell=H}^{s-k+1} A_\ell\right)^c \cap \left(\{L-2d,\ldots,L\}\,\dot\cup\,\{L+1,\ldots,n\}\right)\\
&= \left[\left(\cup_{\ell=H}^{s-k+1} A_\ell\right)^c \cap \{L-2d,\ldots,L\}\right] \dot\cup \left[\left(\cup_{\ell=H}^{s-k+1} A_\ell\right)^c \cap \{L+1,\ldots,n\}\right]\\
&= \left[\left(\cup_{\ell=H}^{s-k+1} A_\ell\right)^c \cap \{L-2d,\ldots,L\}\right] \dot\cup \{L+1,\ldots,n\}
\end{aligned}$$

Since $\left(\cup_{\ell=H}^{s-k+1} A_\ell\right)^c \cap \{L-2d,\ldots,L\}$ and L are part of our sufficient statistics, the above result show that we can construct the action set $\mathcal{D}_s(A_{[H:s-k+1]})$ from the sufficient statistics, which is the desired result.

State Space Sufficiency: Note that

$$\tau(A_{[H:s-k]}) = \left(\sum_{\ell=H}^{s-k}|A_\ell|,\; \hat L,\; \left(\dot\bigcup_{\ell=H}^{s-k} A_\ell\right)^c \cap \{\hat L-2d,\ldots,\hat L\},\; (A_{s-1},\ldots,A_{s-k+1},A_{s-k}),\; \dot\bigcup_{\ell=H}^{s-1} A_\ell\right)$$

where $\hat L=\max\{\ell : \ell\in\dot\bigcup_{\ell=H}^{s-k} A_\ell\}$. Note that $\sum_{\ell=H}^{s-k}|A_\ell| = \sum_{\ell=H}^{s-k-1}|A_\ell| + |A_{s-k}|$ and $\hat L=\max\{L,\max\{\ell : \ell\in A_{s-k}\}\}$, both of which can be computed from the sufficient statistics $\tau(A_{[H:s-k+1]})$ and the action A_{s-k}. Also, note that $(A_{s-1},\ldots,A_{s-k+1},A_{s-k})$ can be computed from A_{s-k} and the sufficient statistics $\tau(A_{[H:s-k+1]})$. Similarly, note that $\dot\bigcup_{\ell=H}^{s-1} A_\ell = \dot\bigcup_{\ell=H}^{s} A_\ell \;\;\dot\cup\, A_{s-1}$, which can be computed from A_{s-1} and the sufficient statistics $\tau(A_{[H:s-k+1]})$.

Thus, it remains to show that

$$\left(\bigcup_{\ell=H}^{s-k} A_\ell\right)^c \cap \{\hat{L}-2d,\ldots,\hat{L}\}$$

can also be computed from the sufficient statistics $\tau(A_{[H:s-k+1]})$ and the action A_{s-k}. Since $\hat{L} = \max\{L, \max\{\ell : \ell \in A_{s-k}\}\}$, there are two cases to consider: $\hat{L}=L$ or $\hat{L}>L$. If $\hat{L}=L$, then

$$\begin{aligned}\left(\bigcup_{\ell=H}^{s-k} A_\ell\right)^c \cap \{\hat{L}-2d,\ldots,\hat{L}\} &= \left(\bigcup_{\ell=H}^{s-k-1} A_\ell\right)^c \cap A_{s-k}^c \cap \{L-2d,\ldots,L\}\\ &= \left[\left(\bigcup_{\ell=H}^{s-k-1} A_\ell\right)^c \cap \{L-2d,\ldots,L\}\right] \cap A_{s-k}^c\end{aligned}$$

which, of course, can be computed from $\tau(A_{[H:s-k+1]})$ and A_{s-k}.

On the other hand, if $\hat{L}>L$, then

$$\left(\bigcup_{\ell=H}^{s-k} A_\ell\right)^c \cap \{\hat{L}-2d,\ldots,\hat{L}\} = \left(\bigcup_{\ell=H}^{s-k-1} A_\ell\right)^c \cap \{\hat{L}-2d,\ldots,\hat{L}\} \cap A_{s-k}^c$$

Since A_{s-k}^c is known (because A_{s-k} is the given action), it suffices to show that $\left(\bigcup_{\ell=H}^{s-k-1} A_\ell\right)^c \cap \{\hat{L}-2d,\ldots,\hat{L}\}$ can be computed from the sufficient statistics $\tau(A_{[H:s-k+1]})$ and A_{s-k}. Note that

$$\begin{aligned}&\left(\bigcup_{\ell=H}^{s-k-1} A_\ell\right)^c \cap \{\hat{L}-2d,\ldots,\hat{L}\}\\ &= \left(\bigcup_{\ell=H}^{s-k-1} A_\ell\right)^c \cap \left[\left(\{L-2d,\ldots,L\} \setminus \{L-2d,\ldots,\hat{L}-2d-1\}\right) \cup \{L+1,\ldots,\hat{L}\}\right]\\ &= \left[\left(\bigcup_{\ell=H}^{s-k-1} A_\ell\right)^c \cap \left(\{L-2d,\ldots,L\} \setminus \{L-2d,\ldots,\hat{L}-2d-1\}\right)\right] \cup \left[\left(\bigcup_{\ell=H}^{s-k-1} A_\ell\right)^c \cap \{L+1,\ldots,\hat{L}\}\right]\\ &= \left[\left(\bigcup_{\ell=H}^{s-k-1} A_\ell\right)^c \cap \{L-2d,\ldots,L\} \cap \{L-2d,\ldots,\hat{L}-2d-1\}^c\right] \cup \{L+1,\ldots,\hat{L}\}\ ,\end{aligned}$$

where the last equality follows from the fact that $X \setminus Y = X \cap Y^c$ and the fact that $\cup_{\ell=H}^{s-k+1} A_\ell \subseteq \{1,\ldots,L\}$, which implies that $(\cup_{\ell=H}^{s-k+1} A_\ell)^c \supseteq \{L+1, L+2,\ldots,n\}$. Note that $\left(\bigcup_{\ell=H}^{s-k-1} A_\ell\right)^c \cap \{L-2d,\ldots,L\}$ is part of the sufficient statistics, and $\hat{L}$ can be computed form the sufficient statistics and A_{s-k}. So, $\left(\bigcup_{\ell=H}^{s-k-1} A_\ell\right)^c \cap \{\hat{L}-2d,\ldots,\hat{L}\}$ can be computed from $\tau(A_{[H:s-k+1]})$ and A_{s-k}, which gives the desired result. ■

It follows from the above discussion that the sufficient statistics for the dynamic programming in Lemma C.4 is given by

$$\tau(A_{[H:s-k+1]}) = \left(\sum_{\ell=H}^{s-k+1} |A_\ell|,\ L,\ \left(\bigcup_{\ell=H}^{s-k+1} A_\ell\right)^c \cap \{L-2d, L-2d+1,\ldots,L\},\ A_{[s:s-k+1]}\right).$$

Note that the first two arguments are numbers in $\{1,\ldots,n\}$; the third argument is a subset of $\{L-2d,\ldots,L\}$. The last argument is a k-dimensional vector $(A_s,\ldots,A_{s-k+1})$. The following lemma bounds the state space by showing that each A_s can be represented as a union of at most $2d+1$ disjoint intervals. To facilitate our exposition, let $[i,j) = \{i, i+1,\ldots,j-1\}$. If $j \le i$, then $[i,j) = \varnothing$.

LEMMA C.6 **(Bound on the State Space)**. *For any $(A_H, A_{H-1}, \ldots, A_1, A_0) \in \mathscr{P}_{d,\alpha}$ and for any $0 \le s \le \ell$, A_s can be written as a union of at most $2d+1$ non-contiguous non-empty intervals; that is, for all s,*

$$A_s = \cup_{h=1}^{q_s} [x_{s,h}, y_{s,h}) \ .$$

where $q_s \le 2d+1$, and

$$x_{s,1} < y_{s,1} < x_{s,2} < y_{s,2} < \cdots < x_{s,q_s} < y_{s,q_s} \quad .$$

Consequently, the number of distinct A_s is at most $\sum_{h=0}^{2d+1} \binom{n}{2h} = O\left(n^{4(d+1)}\right)$.

Proof: There are two cases to consider: $s \ge 1$ and $s = 0$. Let us first consider the case where $s \ge 1$. It is clear that A_s can always written as a union of non-continguous non-empty intervals. It thus suffices to show that there can be at most $2d+1$ such intervals. We will prove this by contradiction. Suppose, on the contrary, that A_s is equal to the union of $2d+2$ non-contiguous and nonempty intervals; that is,

$$A_s = \cup_{h=1}^{2d+1} [x_h, y_h) \ ,$$

where $x_1 < y_1 < x_2 < y_2 < x_3 < y_3 < \cdots < x_{2d+1} < y_{2d+1} < x_{2d+2} < y_{2d+2}$. Note that for all $\ell \ge 2$,

$$\boldsymbol{\pi}^{-1}(x_\ell) = \boldsymbol{\pi}^{-1}(x_{\ell-1}) + (y_{\ell-1} - x_{\ell-1}) \ ,$$

and thus, $\boldsymbol{\pi}^{-1}(x_\ell) = \boldsymbol{\pi}^{-1}(x_1) + \sum_{h=1}^{\ell-1}(y_h - x_h)$. Then,

$$\begin{aligned} x_{2d+2} - \boldsymbol{\pi}^{-1}(x_{2d+2}) &= x_{2d+1} - \sum_{i=1}^{2d+1} (y_i - x_i) - \boldsymbol{\pi}^{-1}(x_1) \\ &= x_1 + \sum_{i=1}^{2d+1} (x_{i+1} - x_i) - \sum_{i=1}^{2d-1} (y_i - x_i) - \boldsymbol{\pi}^{-1}(x_1) \\ &= x_1 + \sum_{i=1}^{2d+1} (x_{i+1} - y_i) - \boldsymbol{\pi}^{-1}(x_1) \\ &\ge x_1 + 2d + 1 - \boldsymbol{\pi}^{-1}(x_1) \ge 2d + 1 - d = d + 1 \ , \end{aligned}$$

where the last inequality follows from the definition of d-sorted constraint, which implies that $-d \le x_1 - \boldsymbol{\pi}^{-1}(x_1) \le d$. However, this contradicts our assumption that $(A_0, \ldots, A_\ell)$ satisfies the d-sorted constraint! Therefore, A_s can be written as a union of at most $2d+1$ non-contiguous non-empty intervals. To bound the number of distinct A_s, consider the union of h disjoint non-contiguous non-empty intervals. Each such union is represented by

$$x_1 < y_1 < x_2 < y_2 < \cdots < x_h < y_h \ ,$$

and this is equivalent of choosing $2h$ distinct numbers from $\{1, 2, \ldots, n\}$. There are $\binom{n}{2h}$ such subsets. So, the total number of distinct A_s is $\sum_{h=0}^{2d+1} \binom{n}{2h} = O(n^{4(d+1)})$.

The above argument makes use of the fact that for any product $i \in A_s$, $|i - \boldsymbol{\pi}^{-1}(i)| \le d$ by the d-sorted constraint. However, this argument does not apply when we consider A_0 because the definition of the d-sorted only enforce the ranking constraint on products that are offered to the customers. So, the case of $s = 0$

requires a separate argument. We claim A_0 can be written as a union of at most $d+1$ non-contiguous non-empty intervals. To see this, suppose on the contrary, that A_0 is equal to the union of $d+2$ non-contiguous and nonempty intervals; that is,

$$A_0 = \cup_{h=1}^{d+2}[x_h, y_h) \ ,$$

where $x_1 < y_1 < x_2 < y_2 < \cdots < x_{d+1} < y_{d+1} < x_{d+2} < y_{d+2}$. Let $\bar{A}_0 = \{1,\ldots,n\} \setminus A_0$, and let h denote the product in $\bar{A}_0$ with the largest index; that is, $h = \max\{i : i \in \bar{A}_0\}$. Note that, by our construction, $y_{d+1} \le h$.

We claim that $h \ge |\bar{A}_0| + d + 1$. To see this, let $S = \{x_1, x_2, \ldots, x_d, x_{d+1}\} \subseteq A_0$. Note that S and $\bar{A}_0$ are disjoint. Since h is the product in $\bar{A}_0$ with the largest index, we have that $h \ge y_{d+1} > \max\{x_1, x_2, \ldots, x_d, x_{d+1}\}$. This implies that $S \uplus \bar{A}_0 \subseteq \{1, 2, \ldots, h\}$. Therefore, $h \ge |S \uplus \bar{A}_0| = |\bar{A}_0| + d + 1$, which establishes the desired claim.

Since all products in $\bar{A}_0$ has a price rank that is smaller than the products in A_0, it follows that $\boldsymbol{\pi}^{-1}(h) \le |\bar{A}_0|$. Moreover, h must satisfies the d-sorted price constraint, so

$$d \ge h - \boldsymbol{\pi}^{-1}(h) \ge h - |\bar{A}_0| \ge \left(|\bar{A}_0| + d + 1\right) - |\bar{A}_0| = d + 1 \ ,$$

but this is a contradiction! Therefore, it must be the case that A_0 can be written as a union of at most $d+1$ non-continguous non-empty intervals. ■

Lemma C.7 **(Bound on the number of possible consideration sets)**. *When product prices are restrticted to be in $\mathscr{P}_{d,\alpha}$, the total number of possible consideration sets is bounded above as follows:*

$$\left| \left\{ \biguplus_{\ell=H}^{s} A_\ell : 1 \le s \le H, (A_H, A_{H-1}, \ldots, A_1, A_0) \in \mathscr{P}_{d,\alpha} \right\} \right| \le \binom{2d}{d} n \le 4^d n$$

Proof: Note that for any price vector $\boldsymbol{p}$, there are at most n possible consideration sets, given by $\left\{\{\pi_{\boldsymbol{p}}(1), \pi_{\boldsymbol{p}}(2), \ldots, \pi_{\boldsymbol{p}}(s)\} : p_{\pi_{\boldsymbol{p}}(s)} < 1, 1 \le s \le n\right\}$, where recall that $\pi_{\boldsymbol{p}}(s)$ is the product with price rank s such that $p_{\pi_{\boldsymbol{p}}(1)} \le p_{\pi_{\boldsymbol{p}}(2)} \le \cdots \le p_{\pi_{\boldsymbol{p}}(n)}$. Because $\mathscr{P}_{d,\alpha} \subseteq \mathscr{P}_d$, the number of possible consideration sets when prices are retricted to belong to $\mathscr{P}_{d,\alpha}$ is bounded above by $|\mathcal{C}_d|$, defined by

$$\mathcal{C}_d = \left\{ \{\pi_{\boldsymbol{p}}(1), \pi_{\boldsymbol{p}}(2), \ldots, \pi_{\boldsymbol{p}}(s)\} : p_{\pi_{\boldsymbol{p}}(s)} < 1, \quad 1 \le s \le n, \quad \boldsymbol{p} \in \mathscr{P}_d, \right\},$$

the collection of all possible consideration sets when prices are restricted to belong to $\mathscr{P}_d$. For that, we upper bound the number of consideration sets of size s for some $1 \le s \le n$.

For an arbitrary price ordering $\pi_{\boldsymbol{p}}$ for $\boldsymbol{p} \in \mathscr{P}_d$, there is at most one consideration set of size s, namely $\{\pi_{\boldsymbol{p}}(1), \pi_{\boldsymbol{p}}(2), \ldots, \pi_{\boldsymbol{p}}(s)\}$, provided $p_{\pi_{\boldsymbol{p}}(s)} < 1$. Because $\boldsymbol{p}$ is d-sorted, it follows by definition that $|\pi_{\boldsymbol{p}}^{-1}(i) - i| \le d$ for $1 \le i$, where recall that $\pi_{\boldsymbol{p}}^{-1}(i)$ is the price rank of product i. Therefore, all the products $1 \le i \le s - d$ must have price ranks $\pi_{\boldsymbol{p}}^{-1}(i) \le i + d \le s$, which implies that $\{1, 2, \ldots, s-d\} \subseteq \{\pi_{\boldsymbol{p}}(1), \pi_{\boldsymbol{p}}(2), \ldots, \pi_{\boldsymbol{p}}(s)\}$, the set of all products with price ranks less than or equal to s. In a similar fashion, $i - d \le \pi_{\boldsymbol{p}}^{-1}(i) \le i + d$ implies that $\pi_{\boldsymbol{p}}(r) - d \le r \le \pi_{\boldsymbol{p}}(r) + d$, which in turn implies that $r - d \le \pi_{\boldsymbol{p}}(r) \le r + d$ for any price rank $1 \le r \le n$. It now follows that $\pi_{\boldsymbol{p}}(r) \le r + d \le s + d$ for any $1 \le r \le s$ and therefore, $\{\pi_{\boldsymbol{p}}(1), \pi_{\boldsymbol{p}}(2), \ldots, \pi_{\boldsymbol{p}}(s)\} \subseteq \{1, 2, \ldots, s+d\}$. We have thus shown that

$$\{1, 2, \ldots, s-d\} \subseteq \{\pi_{\boldsymbol{p}}(1), \pi_{\boldsymbol{p}}(2), \ldots, \pi_{\boldsymbol{p}}(s)\} \subseteq \{1, 2, \ldots, s+d\},$$

where we let $\{\pi_{\boldsymbol{p}}(1), \pi_{\boldsymbol{p}}(2), \ldots, \pi_{\boldsymbol{p}}(s-d)\} = \varnothing$ for $s \le d$.

It now follows that every set $\{\pi_{\boldsymbol{p}}(1), \pi_{\boldsymbol{p}}(2), \ldots, \pi_{\boldsymbol{p}}(s)\}$ is obtained by taking the disjoint union of $\{1, 2, \ldots, s-d\}$ with an arbitrary subset of size d from $\{s-d+1, \ldots, s+d\}$. Because the cardinality of $\{s-d+1, \ldots, s+d\}$ is $2d$, it follows that the number of such sets is bounded above by $\binom{2d}{d}$.

We have thus obtained an upper bound of $\binom{2d}{d}$ for the number of possible consideration sets of size s for each $1 \le s \le n$. Therefore, the total number of possible consideration sets $|\mathcal{C}_d|$ is bounded above by $n\binom{2d}{d}$, which is the desired result.

■

Proof of Proposition 5.1

Proof: It follows from Lemma C.4 that we have the following dynamic programming equation:

$$J_s^*(A_{[H:s-k+1]}) = \max_{A_{s-k} \in \mathcal{D}_s(A_{[H:s-k+1]})} \left\{ \alpha^s G_s(A_{[H:s-k]}) + J_{s-1}^*(A_{[H:s-k]}) \right\} .$$

We have also shown that

$$\tau(A_{[H:s-k+1]}) = \left(\sum_{\ell=H}^{s-k+1} |A_\ell|\,,\ L,\ \left(\dot{\bigcup_{\ell=H}^{s-k+1}} A_\ell \right)^c \cap \{L-2d, L-2d+1, \ldots, L\},\ A_{[s:s-k+1]}\ ,\ \dot{\bigcup_{\ell=H}^{s}} A_\ell \right) .$$

is a sufficient statistics for the above DP. This gives us the following method for computing the value functions.

1. *Initialization.* As a boundary condition, set $J_0^*(\cdot) \equiv 0$.
2. For $s = 1, \ldots, H$ do the following:
   (a) For every feasible value of $\tau(A_{[H:s-k+1]})$, solve the dynamic programming recursion to compute $J_s^*(A_{[H:s-k+1]}) = J_s^*(\tau(A_{[H:s-k+1]}))$ by replacing $A_{[H:s-k+1]}$ with $\tau(A_{[H:s-k+1]})$ in (EC.3). Such replacement is legal because of our definition that $\tau(A_{[H:s-k+1]})$ is a sufficient statistic.
3. Note that for $s = H$, $\tau(A_{[H:s-k+1]})$ is simply $A_{[H:H-k+1]}$. So, given $J_H^*(\cdot)$ for all possible values of $(A_{[H:H-k+1]})$, compute

$$\max_{A_H, \ldots, A_{H-k+1}} J_H^*(A_H, A_{H-1}, \ldots, A_{H-k+1}) = Y_H^* = \max_{\boldsymbol{p} \in \mathscr{P}_{d,\alpha}} R^{\alpha,k}(\boldsymbol{p}) ,$$

   where the last equality follows from Lemma C.3. Store the optimal solution as $A_H^*, A_{H-1}^*, \ldots, A_{H-k+1}^*$.
4. Run backward induction to determine the optimal solution: for $s = H, H-1, \ldots, k$, let

$$A_{s-k}^* = \arg \max_{A_{s-k} \in \mathcal{D}\left(A^*_{[H:s-k+1]}\right)} \left\{ \alpha^s G_s(A^*_{[H:s-k+1]}, A_{s-k}) + J_{s-1}^*(A^*_{[H:s-k+1]}, A_{s-k}) \right\} ,$$

   where $\mathcal{D}\left(A^*_{[H:s-k+1]}\right)$ is the domain associated with $A^*_{[H:s-k+1]}$. If there are multiple optima, pick one solution arbitrarily.
5. The optimal price associated with $Y_H^* = \max_{\boldsymbol{p} \in \mathscr{P}_{d,\alpha}} R^{\alpha,k}(\boldsymbol{p})$ is given by $A_H^*, A_{H-1}^*, \ldots, A_0^*$.

Given $J_{s-1}^*(\cdot)$, computing $J_s^*(\cdot)$ requires us to search over all A_{s-k}. By Lemma C.6, there are at most $O(n^{4(d+1)})$ possible such sets. For each set, we need to compute $G_s(A_{[H:s-k]})$, which takes $O(1)$ by our assumption. So, computing each $J_s^*(\cdot)$ requires $O(n^{4(d+1)})$ operations. The number of possible states of J_s^* is equal to the number of different values of sufficient statistics

$$\tau(A_{[H:s-k+1]}) = \left(\sum_{\ell=H}^{s-k+1} |A_\ell|\,,\ L,\ \left(\dot{\bigcup_{\ell=H}^{s-k+1}} A_\ell \right)^c \cap \{L-2d, L-2d+1, \ldots, L\},\ A_{[s:s-k+1]}\ ,\ \dot{\bigcup_{\ell=H}^{s}} A_\ell \right) .$$

It follows from Lemma C.6 that the number of different values of sufficient statistics is $O\left(n^2 2^{2d+1} n^{4k(d+1)} 4^d n\right) = O\left(n^3 4^{2d} n^{4k(d+1)}\right)$, where the last term follows from the fact that $A_{[s:s-k+1]}$ is a k-dimensional vector and each coordinate has at most $O(n^{4(d+1)})$ values, and by Lemma C.7, the number of distinct $\dot{\bigcup}_{\ell=H}^{s} A_\ell$ is at most $4^d n$. So, the total operations for computing $J_s^*(\cdot)$ is $O\left(n^3 4^{2d} n^{4(k+1)(d+1)}\right)$. We only need to compute up to J_H^*, where $H = O\left(\frac{\log 1/b_{\min}}{\log 1/\alpha}\right)$. This gives the desired result. ■

C.2. Proof of Theorem 5.1

Proof: For any $k \in \mathbb{Z}_{++}$, let

$$\alpha_k = \frac{1}{(2+k)^{\frac{1}{1+k}}}$$

Consider the relaxed revenue function $R^{\alpha_k,k}(\cdot)$. By Proposition C.1, by optimizing this relaxed revenue function, we have a performance guarantee of

$$\alpha_k \left(1 - (\alpha_k)^{k+1}\right) = \frac{1+k}{(2+k)^{1+\frac{1}{1+k}}} .$$

Moreover, note that $k \mapsto \frac{1+k}{(2+k)^{1+\frac{1}{1+k}}}$ is increasing in k because

$$\ln\left(\frac{1+k}{(2+k)^{1+\frac{1}{1+k}}}\right) = \ln\left(\frac{1+k}{2+k}\right) - \frac{\ln(2+k)}{1+k}$$

is increasing in k. In addition,

$$\lim_{k\to\infty} \frac{1+k}{(2+k)^{1+\frac{1}{1+k}}} = 1$$

because

$$\lim_{k\to\infty} \ln\left(\frac{1+k}{(2+k)^{1+\frac{1}{1+k}}}\right) = \lim_{k\to\infty} \ln\left(\frac{1+k}{2+k}\right) - \frac{\ln(2+k)}{1+k} = 0 .$$

Moreover, the running time for solving the dynamic program is

$$O\left(n^3 \times 4^{2d} \times n^{4(k+1)(d+1)} \times \frac{\log 1/b_{\min}}{\log 1/\alpha_k}\right) = O\left(n^3 \times 4^{2d} \times (k+1) \times n^{4(k+1)(d+1)} \times \log \frac{1}{b_{\min}}\right)$$

Let $\epsilon \in (0,1)$ be given, and let

$$k(\epsilon) \stackrel{\text{def}}{=} \left[\frac{4}{\epsilon} + \frac{4}{\epsilon} \ln \frac{1}{\epsilon}\right]^{1\ +\ \frac{1}{\ln \frac{1}{\epsilon}}} .$$

We will show that for any $k \geq k(\epsilon) - 1$, we have

$$\frac{1+k}{(2+k)^{1+\frac{1}{1+k}}} \geq 1 - \epsilon$$

Once we establish the above claim, we simply solve the optimization problem $\max_{\boldsymbol{p} \in \mathscr{P}_{\alpha_k,d}} R^{\alpha_k,k}(\boldsymbol{p})$ using dynamic program. The resulting price vector will have the desired performance guarantee.

Let $u = 1 + k$. Then,

$$\frac{1+k}{(2+k)^{1+\frac{1}{1+k}}} = \frac{u}{1+u} \times \frac{1}{(1+u)^{\frac{1}{u}}} = \left(1 - \frac{1}{1+u}\right) \times e^{-\frac{\ln(1+u)}{u}}$$
$$\geq \left(1 - \frac{\epsilon}{2}\right) \times e^{-\frac{\ln(1+u)}{u}} ,$$

where the last inequality follows from the fact that if $1 + k \geq k(\epsilon)$, then $u \geq \frac{4}{\epsilon}$, and thus,

$$1 - \frac{1}{1+u} \geq 1 - \frac{1}{4/\epsilon} = 1 - \frac{\epsilon}{4} \geq 1 - \frac{\epsilon}{2}$$

It remains to show that $e^{-\frac{\ln(1+u)}{u}} \geq 1 - \frac{\epsilon}{2}$. Note that for all $x > 1$ and all $\delta > 0$,

$$\frac{\ln(1+x)}{x} \leq \frac{(1+x)^\delta - 1}{x\delta} \leq \frac{(1+x)^\delta}{x\delta} \leq \frac{2}{\delta(1+x)^{1-\delta}}$$

Thus, since $u = 1 + k \geq 1$, for all $\delta > 0$,

$$e^{-\frac{\ln(1+u)}{u}} \geq 1 - \frac{\ln(1+u)}{u} \geq 1 - \frac{2}{\delta(1+u)^{1-\delta}}$$

Note that

$$\frac{2}{\delta(1+u)^{1-\delta}} \leq \frac{\epsilon}{2} \quad \Leftrightarrow \quad \frac{4}{\epsilon\delta} \leq (1+u)^{1-\delta} \quad \Leftrightarrow \quad \frac{1}{1-\delta}\left[\ln\frac{4}{\epsilon} + \ln\frac{1}{\delta}\right] \leq \ln(1+u)$$

Pick $\delta > 0$ so that $\frac{1}{\delta} = 1 + \ln\frac{1}{\epsilon}$. Then,

$$\frac{1}{1-\delta}\left[\ln\frac{4}{\epsilon} + \ln\frac{1}{\delta}\right] = \left(1 + \frac{1}{\ln\frac{1}{\epsilon}}\right)\left[\ln\frac{4}{\epsilon} + \ln\left(1 + \ln\frac{1}{\epsilon}\right)\right]$$
$$= \left(1 + \frac{1}{\ln\frac{1}{\epsilon}}\right)\ln\left[\frac{4}{\epsilon} + \frac{4}{\epsilon}\ln\frac{1}{\epsilon}\right] \quad \leq \quad \ln(1+u) ,$$

where the last inequality follow from the definition of u.

Putting everything together, we have that

$$\frac{1+k}{(2+k)^{1+\frac{1}{1+k}}} \geq \left(1 - \frac{\epsilon}{2}\right) \times e^{-\frac{\ln(1+u)}{u}} \geq \left(1 - \frac{\epsilon}{2}\right)\left(1 - \frac{\epsilon}{2}\right) \geq 1 - \epsilon ,$$

which is the desired result. ■

Appendix D: Implementation details for estimating model parameters

We used the algorithm described in Algorithm 1 to initialize the support of the $\lambda^{(0)}$. The objective of the initialization is to find a distribution $\lambda^{(0)}$ that assigns a non-zero probablity to every observation so that the log-likelihood is finite. For that, we start with the 'sales ranking' in which each product is ranked according to its aggregate sales (number of observations in which the product was purchased), with higher sales products having higher ranks. We then obtained $n + 1$ rankings for the support by modifying the sales ranking: ranking i is obtained by moving product i to the top-rank while the remaining products are shifted down in the ranking. Because each product is top-ranked in at least one ranking, every observation has a non-zero probability.

We used the following local search (LS) heuristic in Algorithm 2 to solve the M-step LP. As discussed in Section 4, the M-step LP is equivalent to the following rank aggregation problem: $\arg\max_{\boldsymbol{\sigma} \in \mathscr{S}_{n+1}} \sum_{(a,A) \in \mathcal{C}} c_{a,A} \mathbb{1}[\boldsymbol{\sigma}, a, A]$, with the costs $c_{a,A}$ given. We adopted the following hueristic:

Input $\mathsf{Data} = \{(c_1, S_1, \boldsymbol{p}_1), (c_2, S_2, \boldsymbol{p}_2), \ldots, (c_T, S_T, \boldsymbol{p}_T))\}$; number of products n.

Initialization: Set $n_i \leftarrow \sum_{t=1}^{T} \mathbb{1}[c_t = i]$, i.e., the number of times product i was purchased, for each product i.

Set $\boldsymbol{\sigma}_0$ be th ranking according to the product sales so that $n_{\sigma_i} > n_{\sigma_j}$ if and only if $i < j$; Support $\leftarrow \{\boldsymbol{\sigma}_0\}$.

For $r = 2, 3, \ldots, n+1$:

Define ranking $\boldsymbol{\pi}$ such that $\pi_1 = \sigma_r$, $\pi_j = \sigma_{j-1}$ for $j = 2, \ldots, r$, and $\pi_j = \sigma_j$ for any $j > r$.

Support $\leftarrow$ Support $\cup \{\boldsymbol{\pi}\}$

EndFor

Output. Support

Algorithm 1: Initialization algorithm

Input $c_{a,A}$ for all $(a, A) \in \mathcal{C}$; maximum number of iterations I.

Initialization: Set $\boldsymbol{\sigma}^*$ equal to a permutation sampled uniformly at random from $\mathscr{S}_{n+1}$;

$C(\boldsymbol{\sigma}^*) \leftarrow \sum_{(a,A)\in\mathcal{C}} c_{a,A} \mathbb{1}[\boldsymbol{\sigma}^*, a, A]$

For $i = 1, 2, \ldots, I$:

Neighbors $= \{\boldsymbol{\pi} \in \mathscr{S}_{n+1} : \pi_i = \sigma_j, \pi_j = \sigma_i$ for some $i \neq j\}$ i.e., each neighbor is obtained by interchanging the products ranked at positions i and j for some $i \neq j$.

$C(\boldsymbol{\pi}) \leftarrow \sum_{(a,A)\in\mathcal{C}} c_{a,A} \mathbb{1}[\boldsymbol{\pi}, a, A]$ for each $\boldsymbol{\pi} \in$ Neighbors

$\boldsymbol{\pi}^* = \arg\max_{\boldsymbol{\pi}\in\text{Neighbors}} C(\boldsymbol{\pi})$

If $C(\boldsymbol{\pi}^*) > C(\boldsymbol{\sigma}^*)$, Then

$\boldsymbol{\sigma}^* \leftarrow \boldsymbol{\pi}^*$; $C(\boldsymbol{\sigma}^*) \leftarrow C(\boldsymbol{\pi}^*)$

Else

Break from the For loop because local optimum is reached.

EndIf

EndFor

Output. $\boldsymbol{\sigma}^*$

Algorithm 2: Local search for solving M-step LP

Finally, once the M-step LP was solved, we used the update algorithm in Algorithm 3 to update the distribution $\lambda^{(t)}$ to obtain $\lambda^{(t+1)}$. Given the optimal ranking from the LS heuristic, the algorithm solves a constrained convex program to determine the weights of the rankings in the support that maximize the data log-likelihood. It imposes the constraint that the weights must sum to 1, but does not impose any non-negativity constraints. If the resulting weights are all non-negative, then the algorithm terminates. Otherwise, the ranking with the minimum weight is dropped from the support and the above procedure is repeated. Dropping the rankings allows us to maintain a sparse distribution over rankings.

Appendix E: Implementation details for the optimization simulations

E.1. MILP for joint assortment and price optimization under the nonparametric model

We now describe the mixed integer linear progam (MILP) formulation we used to determine the optimal assortment and price combination under our model. We suppose that the price of each product belongs to

Input: Support(t), the support of $\lambda^{(t)}$; $\boldsymbol{\sigma}^*$ obtained from the LS heuristic; $\hat{\gamma}_{a,A}$ for all $(a,A) \in \mathcal{C}$

Initialization: Set Support$(t+1) \leftarrow$ Support$(t) \cup \{\boldsymbol{\sigma}^*\}$ and $\lambda^{(t+1)}(\boldsymbol{\sigma}) \leftarrow \lambda^{(t)}(\boldsymbol{\sigma})$ for all $\boldsymbol{\sigma} \in$ Support(t) and $\lambda^{(t+1)}(\boldsymbol{\sigma}^*) \leftarrow 0$

While True:

$\boldsymbol{w}^* = \arg\max_{\boldsymbol{w}} \sum_{(a,A)\in\mathcal{C}} \hat{\gamma}_{a,A} \log\left(\sum_{\boldsymbol{\sigma}\in \text{Support}(t)} w_{\boldsymbol{\sigma}} \mathbb{1}[\boldsymbol{\sigma}, a, A]\right)$ subject to $\sum_{\boldsymbol{\sigma}\in\text{Support}(t+1)} w_{\boldsymbol{\sigma}} = 1$,
which is a constrained convex program that can be solved efficiently

If $w^*_{\boldsymbol{\sigma}} < 0$ for some $\boldsymbol{\sigma} \in$ Support$(t+1)$, Then

$\boldsymbol{\pi} \leftarrow \arg\min_{\boldsymbol{\sigma}\in\text{Support}(t+1)} w^*_{\boldsymbol{\sigma}}$

Support$(t+1) \leftarrow$ Support$(t+1) \setminus \{\boldsymbol{\pi}\}$

Else

Break from the While loop because optimal solution is found.

EndIf

EndWhile

Output. Support$(t+1)$; $\lambda^{(t+1)}(\boldsymbol{\sigma}) \leftarrow w^*_{\boldsymbol{\sigma}}$ for all $\boldsymbol{\sigma} \in$ Support$(t+1)$.

Algorithm 3: Update of distribution $\lambda^{(t)}$

one of the L levels $\{b_1, b_2, \ldots, b_L\}$. Without loss of generality, we suppose that $b_\ell \in (0,1)$ for $\ell = 1,2,\ldots,L$. We let $b_0 = 0$ and $b_{L+1} = 1$. We set the price of a product that is not offered at b_{L+1}. The customers draw their price thresholds from the interval $[0, b_L]$. We let g_ℓ denote the probability that the price threshold belongs to the interval $[b_\ell, b_{\ell+1})$ for $\ell = 1,2,\ldots,L$. For notational convenience, we set $g_{L+1} = 0$. We suppose that there are K rankings in the support of λ. Let $\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2, \ldots, \boldsymbol{\sigma}_K$ denote the rankings in the support. We say a customer is of type (ℓ, k) if she has a price threshold in the interval $[b_\ell, b_{\ell+1})$ and preference list $\boldsymbol{\sigma}_k$.

With the above notation, our objective is to find the assortment and price combination that maximizes the expected revenue under our model. For that we introduce the following variables. Let $x_{a\ell}$ denote the indicator that product a is priced at level ℓ for $a \in \mathcal{N}$ and $1 \le \ell \le L$. If $x_{a\ell} = 0$ for all $1 \le \ell \le L$, then we suppose that product a is not offered. Let $y_{a\ell k}$ denote the indicator of whether product a is purchased by the customer of type (ℓ, k) for $a \in \mathcal{N}$, $1 \le \ell \le L$, and $1 \le k \le K$. Finally, let $z_{a\ell k}$ denote the revenue from product a from the customer of type (ℓ, k) for $a \in \mathcal{N}$, $1 \le \ell \le L$, and $1 \le k \le K$.

With these variables, the expected revenue is equal to $\sum_{a\in\mathcal{N}} \sum_{\ell=1}^{L} \sum_{k=1}^{K} g_\ell \lambda(\boldsymbol{\sigma}_k) z_{a\ell k}$. We now discuss the constraints. First, the decision variables $x_{a\ell}$ must satisfy the following constraints:

$$\sum_{\ell=1}^{L} x_{a\ell} \le 1 \; \forall\, a \in \mathcal{N}; \text{ and } x_{a\ell} \in \{0,1\} \text{ for } a \in \mathcal{N}, 1 \le \ell \le L,$$

where the first set of constraints requires that each product is either not offered i.e., $x_{a\ell} = 0$ for all $1 \le \ell \le L$, or is offered at only one price level; and second set of constraints require $x_{a\ell}$ to be binary variables.

We now focus on the constraints for the variables $y_{a\ell k}$:

$$y_{a\ell k} \le \sum_{\ell' \le \ell} x_{a\ell'} \text{ for } a \in \mathcal{N}, 1 \le \ell \le L, 1 \le k \le K$$

$$\sum_{a \in \mathcal{N}} y_{a\ell k} \leq 1 \text{ for } 1 \leq \ell \leq L, 1 \leq k \leq K$$

$$\sum_{a' : \sigma_k(a') > \sigma_k(a)} y_{a'\ell k} \leq 1 - \sum_{\ell' \leq \ell} x_{a\ell'} \text{ for } a \in \mathcal{N}, 1 \leq \ell \leq L, 1 \leq k \leq K$$

$$y_{a\ell k} = 0 \text{ for all } a \in \{a' \in \mathcal{N} : \sigma_k(a') > \sigma_k(0)\}, 1 \leq \ell \leq L, 1 \leq k \leq K$$

$$y_{a\ell k} \in \{0, 1\} \text{ for } a \in \mathcal{N}, 1 \leq \ell \leq L, 1 \leq k \leq K.$$

The first set of constraints impose the requirement that a product must be considered for it to be purchased; specifically, product a will not be purchased by customer of type (ℓ, k) if it is priced at a level strictly above ℓ. The second set of constriaints require the customer of each type to choose at most one product from the set $\mathcal{N}$, which does not include the no-purchase option. The third set of constraints imposes the requirement that each customer chooses the most preferred option from the considered options; in particular, if product a is considered, i.e., priced at a level less than or equal to ℓ, by the customer of type (ℓ, k), then none of the products preferred less than a according to $\boldsymbol{\sigma}_k$ will be purchased. The fourth set of constraints require that a product that is less preferred than the no-purchase option should not be purchased. Finally, the last set of constraints impose the requirement that $y_{a\ell k}$ are binary.

We now consider the set of constraints for $z_{a\ell k}$. It follows from our definitions of variables $x_{a\ell}$ and $y_{a\ell k}$ that the expected revenue from product a from each customer of type (k, ℓ) is equal to 0 if a is not purchased and $b_{\ell'}$ if the a is purchased when priced at level ℓ'. This can be expressed as $z_{a\ell k} = y_{a\ell k} \left(\sum_{\ell' \leq \ell} b_{\ell'} x_\ell \right)$, where the first term captures whether the purchase has happened and the second term captures the expected revenue upon purchase. Because the variable $z_{a\ell k}$ involves product of a binary variable, $y_{a\ell k}$, and a continuous variable, $\sum_{\ell' \leq \ell} b_{\ell'} x_{a\ell'}$, we can replace the product terms with the following set of linear constraints:

$$z_{a\ell k} \leq y_{a\ell k} \text{ for } a \in \mathcal{N}, 1 \leq \ell \leq L, 1 \leq k \leq K$$

$$z_{a\ell k} \leq \sum_{\ell' \leq \ell} b_{\ell'} x_{a\ell'} \text{ for } a \in \mathcal{N} \cup \{0\}, 1 \leq \ell \leq L, 1 \leq k \leq K$$

$$z_{a\ell k} \geq \sum_{\ell' \leq \ell} b_{\ell'} x_{a\ell'} - (1 - y_{a\ell k}) \text{ for } a \in \mathcal{N} \cup \{0\}, 1 \leq \ell \leq L, 1 \leq k \leq K$$

$$z_{a\ell k} \geq 0 \text{ for } a \in \mathcal{N} \cup \{0\}, 1 \leq \ell \leq L, 1 \leq k \leq K.$$

When $y_{a\ell k} = 1$, the second and third constraints require $z_{a\ell k} = \sum_{\ell' \leq \ell} b_{\ell'} x_{a\ell'}$, as desired, and the first constraint, $z_{a\ell k} \leq 1$, is loose (because $b_{\ell'} \in [0, 1]$ for all ℓ'). Similarly, when $y_{a\ell k} = 0$, the first and last constraints require $z_{a\ell k} = 0$, whereas the second and third constraints are loose, again because $\sum_{\ell' \leq \ell} b_{\ell'} x_{a\ell'} \leq 1$. Putting everthing together, we obtain the following mixed integer linear program (MILP):

$$\max_{\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}} \sum_{a \in \mathcal{N}} \sum_{\ell=1}^{L} \sum_{k=1}^{K} g_\ell \lambda(\boldsymbol{\sigma}_k) z_{a\ell k}$$

subject to

$$\sum_{\ell=1}^{L} x_{a\ell} \leq 1 \text{ for } a \in \mathcal{N}, 1 \leq \ell \leq L$$

$$y_{a\ell k} \leq \sum_{\ell' \leq \ell} x_{a\ell'} \text{ for } a \in \mathcal{N}, 1 \leq \ell \leq L, 1 \leq k \leq K$$

$$\sum_{a \in \mathcal{N}} y_{a\ell k} \leq 1 \text{ for } 1 \leq \ell \leq L, 1 \leq k \leq K$$

$$\sum_{a' \colon \sigma_k(a') > \sigma_k(a)} y_{a'\ell k} \leq 1 - \sum_{\ell' \leq \ell} x_{a\ell'} \text{ for } a \in \mathcal{N}, 1 \leq \ell \leq L, 1 \leq k \leq K$$
$$y_{a\ell k} = 0 \text{ for all } a \in \{a' \in \mathcal{N} \colon \sigma_k(a') > \sigma_k(0)\}, 1 \leq \ell \leq L, 1 \leq k \leq K$$
$$z_{a\ell k} \leq y_{a\ell k} \text{ for } a \in \mathcal{N}, 1 \leq \ell \leq L, 1 \leq k \leq K$$
$$z_{a\ell k} \leq \sum_{\ell' \leq \ell} b_{\ell'} x_{a\ell'} \text{ for } a \in \mathcal{N}, 1 \leq \ell \leq L, 1 \leq k \leq K$$
$$z_{a\ell k} \geq \sum_{\ell' \leq \ell} b_{\ell'} x_{a\ell'} - (1 - y_{a\ell k}) \text{ for } a \in \mathcal{N}, 1 \leq \ell \leq L, 1 \leq k \leq K$$
$$y_{a\ell k}, z_{a\ell k} \geq 0 \text{ for } a \in \mathcal{N}, 1 \leq \ell \leq L, 1 \leq k \leq K$$
$$x_{a\ell}, y_{a\ell k} \in \{0,1\} \text{ for } a \in \mathcal{N}, 1 \leq \ell \leq L, 1 \leq k \leq K.$$

The above MILP may be simplified by dropping a few constraints. First, we argue that the constraints $y_{a\ell k} \in \{0,1\}$ and $\sum_{a \in \mathcal{N}} y_{a\ell k} \leq 1$ may be replaced by $0 \leq y_{a\ell k} \leq 1$ for all a, k, ℓ. To see this, consider a feasible price assignment $\boldsymbol{x}$ and type (ℓ, k). If product a is *not* considered or offered, then the second constraint, combined with the non-negativity constraint, forces $y_{a\ell k}$ to be zero. If, on the other hand, product a is considered, then the fourth and fifth constraints force $y_{a\ell k}$ to be zero unless it is the top-ranked product among the considered products according to $\boldsymbol{\sigma}_k$. It now follows that $y_{ak\ell} = 0$ for all $a \neq a^*_{k\ell}$, where $a^*_{\ell k}$ is the most preferred product among the considered products by type (ℓ, k) for price assignment $\boldsymbol{x}$. Now because the objective is to maximize the expected revenues, at optimality, we must have $y_{a^* k\ell k\ell}$ large enough such that $z_{a^*_{k\ell}\ell k} = \sum_{\ell' \leq \ell} b_{\ell'} x_{a^*_{k\ell}\ell'}$. Therefore, replacing the constraints $y_{a\ell k} \in \{0,1\}$ and $\sum_{a \in \mathcal{N}} y_{a\ell k} \leq 1$ by $0 \leq y_{a\ell k} \leq 1$ will not affect the optimal solution. Second, the constraint $\sum_{\ell=1}^{L} x_{a\ell} \leq 1$ is subsumed by the fourth set of constraints with $\ell = L$, $\sum_{\ell'=1}^{L} x_{a\ell'} \leq 1 - \sum_{a' \colon \sigma_k(a') > \sigma_k(a)} y_{a'\ell k} \leq 1$, and hence may be dropped.

With the above simplification, we have the following MILP:

$$\max_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}} \sum_{a \in \mathcal{N}} \sum_{\ell=1}^{L} \sum_{k=1}^{K} g_\ell \lambda(\boldsymbol{\sigma}_k) z_{a\ell k} \qquad \text{(Nonparametric Joint Opt)}$$

subject to

$$y_{a\ell k} \leq \sum_{\ell' \leq \ell} x_{a\ell'} \text{ for } a \in \mathcal{N}, 1 \leq \ell \leq L, 1 \leq k \leq K$$
$$\sum_{a' \colon \sigma_k(a') > \sigma_k(a)} y_{a'\ell k} \leq 1 - \sum_{\ell' \leq \ell} x_{a\ell'} \text{ for } a \in \mathcal{N}, 1 \leq \ell \leq L, 1 \leq k \leq K$$
$$y_{a\ell k} = 0 \text{ for all } a \in \{a' \in \mathcal{N} \colon \sigma_k(a') > \sigma_k(0)\}, 1 \leq \ell \leq L, 1 \leq k \leq K$$
$$z_{a\ell k} \leq y_{a\ell k} \text{ for } a \in \mathcal{N}, 1 \leq \ell \leq L, 1 \leq k \leq K$$
$$z_{a\ell k} \leq \sum_{\ell' \leq \ell} b_{\ell'} x_{a\ell'} \text{ for } a \in \mathcal{N}, 1 \leq \ell \leq L, 1 \leq k \leq K$$
$$z_{a\ell k} \geq \sum_{\ell' \leq \ell} b_{\ell'} x_{a\ell'} - (1 - y_{a\ell k}) \text{ for } a \in \mathcal{N}, 1 \leq \ell \leq L, 1 \leq k \leq K$$
$$z_{a\ell k} \geq 0, 0 \leq y_{a\ell k} \leq 1 \text{ for } a \in \mathcal{N}, 1 \leq \ell \leq L, 1 \leq k \leq K$$
$$x_{a\ell} \in \{0,1\} \text{ for } a \in \mathcal{N}, 1 \leq \ell \leq L, 1 \leq k \leq K.$$

E.2. MILP for joint assortment and price optimization under the LC-MNL model

We formulated the joint assortment and price optimization problem under the LC-MNL model with K classes as an MILP by modifying the MILP proposed in [5, p. 251, equation (7)] for determining the assortment of products that maximize the revenue. To obtain the formulation, consider an LC-MNL model with K classes and suppose that the price of each product is one of the L levels $\{b_1, b_2, \ldots, b_L\}$. Let $v_{ka\ell}$ denote the preference weight for class k, product a, and price level ℓ. The decision variables then are $x_{a\ell} \in \{0,1\}$, indicating whether product a is offered at price level ℓ, and our objective is to set the decision variables such that the expected revenue is maximized. This problem is equivalent to choosing the revenue maximizing assortment of products from nL products, obtained by creating L copies of each product, one for each price level, with the added constraint that at most one copy of each product is chosen. Therefore, the MILP for joint assortment and price optimization can be obtained by adding that at most one copy of each product is chosen to the MILP for choosing the optimal assortment. More precisely, we obtain the following MILP:

$$\max_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}} \sum_{a\in\mathcal{N}} \sum_{\ell=1}^{L} \sum_{k=1}^{K} \alpha_k b_\ell v_{ka\ell} z_{ka\ell} \qquad \text{(LC-MNL JOINT OPT)}$$

$$\begin{aligned}
\text{subject to} \qquad y_k + \sum_{(a,\ell)\in\mathcal{N}_L} v_{ka\ell} z_{ka\ell} &= 1 \text{ for all } 1 \le k \le K \\
y_k - z_{ka\ell} &\le 1 - x_{a,\ell} \text{ for all } 1 \le k \le K, (a,\ell) \in \mathcal{N}_L \\
z_{ka\ell} &\le y_k \text{ for all } 1 \le k \le K, (a,\ell) \in \mathcal{N}_L \\
(1 + v_{ka\ell}) z_{ka\ell} &\le y_k \text{ for all } 1 \le k \le K, (a,\ell) \in \mathcal{N}_L \\
\sum_{\ell=1}^{L} x_{a\ell} &\le 1 \text{ for all } a \in \mathcal{N} \\
x_{a\ell} \in \{0,1\}, y_k, z_{ka\ell} &\ge 0 \text{ for all } 1 \le k \le K, (a,\ell) \in \mathcal{N}_L,
\end{aligned}$$

where $\mathcal{N}_L \stackrel{\text{def}}{=} \{(a,\ell) \colon a \in \mathcal{N}, 1 \le \ell \le L\}$ denotes the universe of nL products, obtained by creating L copies of each product, one for each price level; the penultimate constraint imposes the requirement that only one copy of each product is offered; and the remaining constraints are as described in [5].

Appendix F: EM algorithm for censored demand data with unobserved no-purchase

We now discuss a simple extension of our estimation procedure to account for missing observations when customers leave without purchasing. In many applications, such observations are not recorded as part of sales transactions. Several techniques have been proposed in the literature (see Farias and Jagabathula [2] and Vulcano and van Ryzin [4]) to deal with such missing observations. Existing techniques run the gamut from heuristic extrapolations to fill in the missing entries (Haensel and Koole [3]) to systematic EM-based inference techniques (Vulcano and van Ryzin [4]).

To deal with missing observations in the context of our model, we adopt the approach of Vulcano and van Ryzin [4] and extend their EM-based algorithm to our setting. The setup is as follows. Sales occur over T discrete time periods. In each time period, there is at most one customer arrival. Arrivals occur according to a Bernoulli process with arrival rate $\gamma < 1$. The firm records customer purchases but does not record the periods in which a customer arrives but does not purchase. Therefore, in a period without a purchase, either a customer arrived but decided not to purchase or there was no customer arrival. Let $\mathcal{O} \subseteq \{1, 2, \ldots, T\}$ denote the time periods with purchases. For any $t \in \mathcal{O}$, let $(c_t, S_t, \boldsymbol{p}_t)$ denote the tuple of observations: purchased product c_t when offered the assortment and price combination $S_t, \boldsymbol{p}_t$. To simplify notation, let θ_t denote the choice probability $\theta_{c_t}(S_t, \boldsymbol{p}_t)$ under our model. The incomplete data log-likelihood can now be written as

$$\mathcal{L}_I = \sum_{t \in \mathcal{O}} \log(\gamma\theta_t) + \sum_{t \notin \mathcal{O}} \log\left(\gamma\theta_{0t} + 1 - \gamma\right),$$

where we let θ_{0t} denote the probability of choosing the no purchase alternative under our model from the offer set and price combination $S_t, \boldsymbol{p}_t$, for any $t \notin \mathcal{O}$. The first term in the above expression captures the data log-likelihood in the time periods with purchases and the second term the data log-likelihood in the periods without purchases. The sum $\gamma\theta_{0t} + (1-\gamma)$ captures the fact that a period without a purchase will happen in one of the two ways: arrival but no purchase with probabilty $\gamma\theta_{0t}$ or no arrival with probability $1-\gamma$. The above log-likelihood function is in general hard to maximize. Therefore, we adopt the EM heuristic.

We introduce latent variables $q_t \in \{0,1\}$ indicating whether there was a customer arrival or not in period t. If the latent variables q_t are observed, we obtain the complete data log-likelihood function

$$\mathcal{L}_C = \sum_{t \in \mathcal{O}} \log(\gamma\theta_t) + \sum_{t \notin \mathcal{O}} \left(\mathbb{1}[q_t = 1]\log\gamma\theta_{0t} + \mathbb{1}[q_t = 0]\log(1-\gamma)\right)$$

With q_t observed, it is clear that the complete data log-likelihood is separable in γ and the parameters defining our model θ_t. Therefore, estimating the paramters requires us to solve the following optimization problems:

$$\max_{0 \le \gamma \le 1} \left(\sum_t \mathbb{1}[q_t = 1]\right)\log\gamma + (\mathbb{1}[q_t = 0])\log(1-\gamma) \quad \text{and} \quad \max_{g,\lambda} \sum_{t \in \mathcal{O}} \log\theta_t + \sum_{t \notin \mathcal{O}:\; q_t = 1} \log\theta_{0t}. \tag{EC.4}$$

Optimizing the first problem results in the optimal solution $\hat{\gamma} = (1/T)\sum_t \mathbb{1}[q_t = 1]$, the fraction of time periods in which arrivals occured. The second optimization problem is the same as the MLE problem in (1). Therefore, we can solve this problem using the EM algorithm proposed in Section 3.

Of course, the latent variables q_t for $t \notin \mathcal{O}$ are not known. Therefore, we compute the conditional expected values: for all $t \notin \mathcal{O}$,

$$\begin{aligned}\hat{q}_t = \mathbb{E}\left[q_t | t \notin \mathcal{O}, \gamma, \boldsymbol{\theta}\right] = \Pr(q_t = 1 | t \notin \mathcal{O}, \gamma, \boldsymbol{\theta}) &= \frac{\Pr(t \notin \mathcal{O} | q_t = 1, \gamma, \boldsymbol{\theta})\Pr(q_t = 1|\gamma, \boldsymbol{\theta})}{\Pr(t \notin \mathcal{O}|\gamma, \boldsymbol{\theta})} \\ &= \frac{\theta_{0t}\gamma}{\gamma\theta_{0t} + 1 - \gamma},\end{aligned} \tag{EC.5}$$

where $\boldsymbol{\theta}$ denotes the vector of choice probabilities $((\theta_t)_{t \in \mathcal{O}}, (\theta_{0t})_{t \notin \mathcal{O}})$. It follows by definition that $\hat{q}_t = 1$ for all $t \in \mathcal{O}$.

The above two steps are performed iteratively until convergence. Specifically, we start with initial estimates of $\boldsymbol{\theta}$ and γ. Given the initial estimates, we infer the values $\hat{q}_t$ using (EC.5) (the E-step). We then use the estimates $\hat{q}_t$ to solve the optimization problems in (EC.4) (the M-step). The first problem in (EC.4) can be solved in closed form but the second problem requires running the EM algorithm in Section 3. It follows from standard EM machinery that carrying out the above procedure is always guaranteed to converge to a stationary point (see Vulcano and van Ryzin [4]).

Appendix G: PTAS for Constrained Joint Assortment and Price Optimization

The goal of this section is to develop a PTAS for the constrained joint assortment and price optimization problem. As in the unconstrained case, throughout this section, we assume that the products are indexed by $1,\ldots,n$, and the reference rank of product i is i; that is $\boldsymbol{\tau}$ is the identity ordering. Also, recall that we assume that $\max_{b\in\mathcal{B}} b<1$, and thus, pricing a product at 1 effectively removes it from the offer set. Therefore, the d-sorted family of prices is given by

$$\mathscr{P}_d=\left\{\boldsymbol{p}\in[0,1]^n \;:\; \max_{i\,:\,p_i<1}\left|\pi_{\boldsymbol{p}}^{-1}(i)-i\right|\le d\right\}$$

where $\pi_{\boldsymbol{p}}$ represents the price ordering under $\boldsymbol{p}$, with $p_{\pi_{\boldsymbol{p}}(1)}\le p_{\pi_{\boldsymbol{p}}(2)}\le\cdots\le p_{\pi_{\boldsymbol{p}}(n)}$, and for any i, $\pi_{\boldsymbol{p}}(i)$ denotes the product at rank i under $\boldsymbol{p}$. Note that $\pi_{\boldsymbol{p}}^{-1}(i)$ denote the price rank of product i.

We will focus on the following constrained optimization problem:

$$Z_\alpha^*=\max_{\boldsymbol{p}\in\mathscr{P}_d\cap\mathsf{Box}_\alpha} R(\boldsymbol{p})$$

where $\mathsf{Box}_\alpha=\{\boldsymbol{p}\in[0,1]^n:\alpha^{L_i}\le p_i\le\alpha^{U_i}\quad\forall\ i\}$, and $\alpha\in(0,1)$ and $U_i,L_i\in\mathbb{Z}_+$.

Recall that $\mathscr{P}_{d,\alpha}\stackrel{\text{def}}{=}\{\boldsymbol{p}\in\mathscr{P}_d:\,p_i\in\mathsf{Dom}_\alpha\ \forall\ i\}$. The following lemma shows that by restricting our search to a discrete set $\mathscr{P}_{d,\alpha}$, the maximum revenue decreases by at most a factor of α.

Lemma G.1 **(Rounding)**. *For any $\alpha\in(0,1)$ and $\boldsymbol{p}\in\mathscr{P}_d\cap\mathit{Box}_\alpha$, there exists $\hat{\boldsymbol{p}}\in\mathscr{P}_{d,\alpha}\cap\mathit{Box}_\alpha$ such that $R(\hat{\boldsymbol{p}})\ \ge\ \alpha R(\boldsymbol{p})$. Consequently, $\max_{\boldsymbol{p}\in\mathscr{P}_{d,\alpha}\cap\mathit{Box}_\alpha} R(\boldsymbol{p})\ge\alpha Z_\alpha^*$.*

Proof: For all i, set $\hat{p}_i=\max\{x\in\mathsf{Dom}_\alpha:x\le p_i\}$. Since $\boldsymbol{p}\in\mathscr{P}_d\cap\mathsf{Box}_\alpha$, it follows that $\hat{\boldsymbol{p}}\in\mathscr{P}_{d,\alpha}\cap\mathsf{Box}_\alpha$. The rest of the proof is exact the same as the one for Lemma C.1 in Appendix C, and we omit the details. ■

As in the unconstrained case, we define the relaxed revenue function

$$R^{\alpha,k}(\boldsymbol{p})\stackrel{\text{def}}{=}\sum_{(b,\boldsymbol{\sigma})} w(b,\boldsymbol{\sigma})\sum_{i=1}^n \mathbb{1}\left[\boldsymbol{\sigma},i,\left\{\ell:p_\ell\le b\text{ and }p_\ell\le p_i/\alpha^k\right\}\right]\ .$$

The next proposition relates the solution of the relaxed revenue function to the solution of the original revenue function. The proof of this result is exactly the same as the proof for the unconstrained case, and we omit the details.

Proposition G.1 **(Approximation Guarantee)**. *For any $\alpha\in(0,1)$ and $k\in\mathbb{Z}_{++}$, let $\boldsymbol{p}^{\alpha,k}$ be an optimal solution to the optimization problem associated with the <u>relaxed</u> revenue function $\max_{\boldsymbol{p}\in\mathscr{P}_{d,\alpha}\cap\mathit{Box}_\alpha} R^{\alpha,k}(\boldsymbol{p})$. Then,*

$$Z_\alpha^*\ge R(\boldsymbol{p}^{\alpha,k})\ge\alpha(1-\alpha^{k+1})Z_\alpha^*\ .$$

The main result of this section is to show that the constrained optimization problem associated with the relaxed revenue function can be solved efficiently in polynomial time.

Proposition G.2 **(DP for the Relaxed Problem)**. *The problem $\max_{\boldsymbol{p}\in\mathscr{P}_{d,\alpha}\cap\mathit{Box}_\alpha} R^{\alpha,k}(\boldsymbol{p})$ can be solved via dynamic programming with a running time of $O\left(n^3\times 4^{2d}\times n^{4(k+1)(d+1)}\times\frac{\log 1/b_{\min}}{\log 1/\alpha}\right)$.*

G.1. Proof of Proposition G.2

It turns out that the solution to the optimization $\max_{\boldsymbol{p}\in\mathscr{P}_{d,\alpha}\cap\mathtt{Box}_\alpha} R^{\alpha,k}(\boldsymbol{p})$ can be obtained using dynamic programming method that is very similar to the unconstrained case. We only need to modify the DP equation slightly. So, we will give a brief overview of the changes.

An Equivalent Representation of Prices: As before, let α^H denote the largest price that is less than or equal to the smallest budget; that is, $\alpha^H = \max\{\alpha^s : \alpha^s \le \min_{b\in\mathcal{B}} b\}$, or equivalently, $H = \min\{s : \alpha^s \le \min_{b\in\mathcal{B}} b\}$. By our construction, it is never optimal to consider prices less than α^H. Also, let

$$\mathsf{Valid}_s = \left\{ i \in \{1,2,\ldots,n\} \ : \ \alpha^{L_i} \le \alpha^s \le \alpha^{U_i} \right\} \ ,$$

denote the set of all products that can be priced at α^s. Then, we can represent each price vector $\boldsymbol{p} \in \mathscr{P}_{d,\alpha} \cap \mathtt{Box}_\alpha$ can be represented as a vector $(A_H, A_{H-1}, \ldots, A_1, A_0)$, where for all s,

$$A_s = \{ i \in \mathsf{Valid}_s : p_i = \alpha^s \}$$

is the set of products whose prices are equal to α^s. Throughout this section, we will use this representation of prices. Note that it is possible that $A_s = \varnothing$ for some s. Also, since the maximum budget is less than one, $A_0 = \{i : p_i = \alpha^0 = 1\}$ effectively correspond to the set of products that we will *not* offer. Note that this is virtually the same as in the unconstrained case, and the only difference is in the introduction of the set Valid_s, which will ensure that the price vector satisfies the box constraints.

An Equivalent Optimization Problem: As before, we can rephrase the optimization problem associated with $R^{\alpha,k}$ as follows. For any $(A_H, A_{H-1}, \ldots, A_1, A_0)$ and $0 \le s \le H$, let $W_s^{\alpha,k}(A_H, A_{H-1}, \ldots, A_1, A_0)$ denote the total revenue under the relaxed revenue function that is collected from the products in $A_s \cup A_{s-1} \cup \cdots A_1 \cup A_0$; that is,

$$\begin{aligned}
&W_s^{\alpha,k}(A_H, A_{H-1}, \ldots, A_1, A_0) \\
&\stackrel{\text{def}}{=} \sum_{(b,\boldsymbol{\sigma})} w(b,\boldsymbol{\sigma}) \sum_{\ell=1}^{s} \alpha^\ell \sum_{i\in A_\ell} \mathbb{1}\left[\boldsymbol{\sigma}, i, \left\{ q \in \{0\} \cup A_H \cup A_{H-1} \cup \cdots \cup A_1 \cup A_0 : p_q \le b \text{ and } p_q \le \frac{p_i}{\alpha^k} \right\}\right] \\
&= \sum_{(b,\boldsymbol{\sigma})} w(b,\boldsymbol{\sigma}) \sum_{\ell=1}^{s} \alpha^\ell \sum_{i\in A_\ell} \mathbb{1}\left[\boldsymbol{\sigma}, i, \{ q \in \{0\} \cup A_H \cup A_{H-1} \cup \cdots \cup A_{\ell-k} : p_q \le b \}\right]
\end{aligned}$$

Note that since $b < 1$ for all $b \in \mathcal{B}$, products in A_0 are never selected. For $s = 1, \ldots, H$, let

$$Y_{s,\alpha}^* \stackrel{\text{def}}{=} \max_{(A_H, A_{H-1}, \ldots, A_1, A_0) \in \mathscr{P}_{d,\alpha} \cap \mathtt{Box}_\alpha} W_s^{\alpha,k}(A_H, A_{H-1}, \ldots, A_1, A_0)$$

As before, the following lemma shows that $Y_{H,\alpha}^*$ is our desired target.

Lemma G.2 **(Equivalent Optimization)**. $Y_{H,\alpha}^* = \max_{\boldsymbol{p}\in\mathscr{P}_{d,\alpha}\cap \mathit{Box}_\alpha} R^{\alpha,k}(\boldsymbol{p})$.

Based on Lemma C.3, we will develop dynamic programming methods for computing $Y_0^*, Y_1^*, \ldots, Y_H^*$. Define the following the value functions. For $s = 0, 1, \ldots, H$, let

$$
\begin{aligned}
& J_s^*(A_H, A_{H-1}, \ldots, A_s, A_{s-1}, \ldots, A_{s-k+2}, A_{s-k+1}) \\
& \stackrel{\text{def}}{=} \quad \max_{(A_{s-k}, \ldots, A_0)} \Big\{ W_s^{\alpha,k}(A_H, A_{H-1}, \ldots, A_s, \ldots, A_{s-k+2}, A_{s-k+1}, A_{s-k}, \ldots, A_0) \;\Big| \\
& \qquad\qquad (A_H, A_{H-1}, \ldots, A_{s-k+1}, A_{s-k}, \ldots, A_0) \in \mathscr{P}_{d,\alpha} \cap \mathtt{Box}_\alpha \Big\}
\end{aligned}
$$

Note that

$$
\max_{(A_H, \ldots, A_{s-k+2}, A_{s-k+1})} J_s^*(A_H, A_{H-1}, \ldots, A_{s-k+2}, A_{s-k+1}) = \max_{(A_H, A_{H-1}, \ldots, A_1, A_0) \in \mathscr{P}_{d,\alpha} \cap \mathtt{Box}_\alpha} W_s^{\alpha,k}(A_H, A_{H-1}, \ldots, A_1, A_0) = Y_{s,\alpha}^*
$$

The following lemma provides a dynamic programming recursion for computing the value function.

LEMMA G.3 **(Dynamic Programming Equation)**. *For any s,*

$$
\begin{aligned}
& J_s^*(A_H, A_{H-1}, \ldots, A_s, A_{s-1}, \ldots, A_{s-k+2}, A_{s-k+1}) \\
& = \max_{A_{s-k} \in \mathcal{D}_s(A_H, A_{H-1}, \ldots, A_{s-k+2}, A_{s-k+1})} \Big\{ \alpha^s G_s(A_H, \ldots, A_{s-1}, A_s, \ldots, A_{s-k+1}, A_{s-k}) \\
& \qquad\qquad + J_{s-1}^*(A_H, A_{H-1}, \ldots, A_s, A_{s-1}, A_{s-2}, \ldots, A_{s-k+1}, A_{s-k}) \Big\},
\end{aligned}
$$

where

$$
G_s(A_H, \ldots, A_{s-1}, A_s, \ldots, A_{s-k+1}, A_{s-k}) = \sum_{(b,\boldsymbol{\sigma})} w(b, \boldsymbol{\sigma}) \sum_{i \in A_s} \mathbb{1}\left[\boldsymbol{\sigma}, i, \{q \in \{0\} \cup A_H \cup A_{H-1} \cup \cdots \cup A_{s-k} : p_q \le b\}\right],
$$

and the set $\mathcal{D}_s(A_H, A_{H-1}, \ldots, A_{s-k+2}, A_{s-k+1})$ denotes the collection of subset of products that can be priced at α^{s-k} and still satisfy the d-sorted *constraint; that is,*

$$
\begin{aligned}
& \mathcal{D}_s(A_H, A_{H-1}, \ldots, A_{s-k+2}, A_{s-k+1}) \\
& \stackrel{def}{=} \left\{ X \subseteq \left(\cup_{\ell=H}^{s-k+1} A_\ell\right)^c \cap \mathsf{Valid}_{s-k} \;:\; \max_{\ell \in X} \left|\boldsymbol{\pi}^{-1}(\ell) - \ell\right| \le d \right\} \\
& = \left\{ X \;:\; X = \{i_1, \ldots, i_q\} \subseteq \left(\cup_{\ell=H}^{s-k+1} A_\ell\right)^c \cap \mathsf{Valid}_{s-k} \text{ for some } i_1 < i_2 < \cdots < i_q, \text{ and } \max_{u=1,\ldots,q} \left| \left(\sum_{\ell=H}^{s-k+1} |A_\ell| + u \right) - i_u \right| \le d \right\},
\end{aligned}
$$

where the last equality follows from the fact that the rank of the first product in A_{s-k} is equal to $1 + \sum_{\ell=H}^{s-k+1} |A_\ell|$.

We note that the the above dynamic programming equation is exactly the same as in the unconstrained case. The only difference here is that the requiring that the products that will be priced at α^{s-k} must be valid products; that is, they must satisfy the box constraints, or equivalently, the products must be from the set Valid_{s-k}.

Given that we essentially the same DP formulation as in the unconstrained, using exactly the same argument, we can show that

$$
\tau(A_{[H:s-k+1]}) = \left(\sum_{\ell=H}^{s-k+1} |A_\ell|,\; L,\; \left(\dot{\bigcup_{\ell=H}^{s-k+1}} A_\ell \right)^c \cap \{L-2d, L-2d+1, \ldots, L\},\; A_{[s:s-k+1]},\; \dot{\bigcup_{\ell=H}^{s}} A_\ell \right),
$$

is a sufficient statistics for the dynamic program. Then, we get the desired result.

References

[1] A. Ali and M. Meliă. Experiments with Kemeny ranking: What works when? *Mathematical Social Sciences*, 64:28–40, 2012.

[2] V. Farias, S. Jagabathula, and D. Shah. A new approach to modeling choice with limited data. *Management Science*, 59(2):305–322, 2013.

[3] A. Haensel and G. Koole. Estimating unconstrained demand rate functions using customer choice sets. *Journal of Revenue and Pricing Management*, 10(5):438–454, 2011.

[4] Garrett van Ryzin and Gustavo Vulcano. An expectation-maximization algorithm to estimate a general class of nonparametric choice models. *NYU Stern Working Paper*, 2013.

[5] I. Mendez-Diaz, J. Miranda Bront, G. Vulcano, and P. Zabala. A branch-and-cut algorithm for the latent-class logit assortment problem. Working Paper, New York University, 2010.