The Exponomial Choice Model for Assortment Optimization: An Alternative to the MNL Model?

Ali Aouad
London Business School, London NW1 4SA, United Kingdom. aouad.ma@gmail.com

Jacob Feldman
Olin Business School, Washington University, 1 Brookings Dr., St. Louis, Missouri 63130, USA. jbfeldman@wustl.edu

Danny Segev
Department of Statistics and Operations Research, Tel Aviv University, Tel Aviv 69978, Israel. segevdanny@tauex.tau.ac.il

In this paper, we consider the yet-uncharted assortment optimization problem under the Exponomial choice model, where the objective is to determine the revenue maximizing set of products that should be offered to customers. Our main algorithmic contribution comes in the form of a fully polynomial-time approximation scheme (FPTAS), showing that the optimal expected revenue can be efficiently approached within any degree of accuracy. We synthesize several ideas related to approximate dynamic programming, intended to construct a compact discretization of the continuous state space by keeping track of "key statistics" in rounded form and by operating with a suitable bit precision complexity. We complement this result by a number of NP-hardness reductions to natural extensions of this problem. Moreover, we conduct empirical and computational evaluations of the Exponomial choice model and our solution method. Focusing on choice models with a simple parametric structure, we provide new empirical evidence that the Exponomial choice model can achieve higher predictive accuracy than the Multinomial Logit (MNL) choice model on several real-world data sets. We uncover that this predictive performance correlates with certain characteristics of the choice instance – namely, the entropy and magnitude of choice probabilities. Finally, we leverage fully ranked preference data to simulate the expected revenue of optimal assortments prescribed using the fitted Exponomial and MNL models. On semi-synthetic data, the Exponomial-based approach can lift revenues by 3-4% on average against the corresponding MNL benchmark.

Key words: Assortment Optimization, FPTAS, Approximate Dynamic Programming, Case Study.

1. Introduction

Assortment optimization has come to be one of the most fundamental and well-studied computational questions in revenue management. In this setting, a retailer wishes to determine which set of products (or assortment) should be offered to arriving customers, with the objective of maximizing expected revenue. The foremost challenge in such problems is that of carefully balancing the appeal of the offered assortment to customers and its profitability to the retailer. Adding any product to a given assortment diversifies the retailer's display and thus increases its market share; however, this newly added product can potentially cannibalize the sales of already-offered products.

Modeling choice. The exact nature of this trade-off is determined by the underlying choice model, which is specified by the retailer, and then used to determine the probability that each product is purchased within a given assortment. In the last few decades, a variety of customer choice models have been explored and exploited by the economics, marketing, and operations management communities, each of which captures different aspects of customer substitution behavior. We refer the reader to the excellent book of Train (2009) for a detailed overview of this topic.

From a practical perspective, in order to formulate a concrete assortment optimization problem, one is initially required to pick an appropriate customer choice model and to estimate its parameters from historical data. Model selection is often a challenging task, due to the inherent tension between model accuracy and computational tractability. Indeed, choice models that capture general and fine-grained forms of customer behavior are precisely those whose estimation methods are the most computationally intensive and whose corresponding assortment optimization problems are notoriously hard to solve. For this reason, the Multinomial Logit (MNL) model, originally pioneered by Luce (1959) and McFadden (1974), has become the gold standard for customer choice models due to the ease with which one can derive accurate parameter estimates (Train 2009, Vulcano et al. 2012) and due to the tractability of its accompanying assortment optimization problems (Talluri and van Ryzin 2004, Rusmevichientong et al. 2010, Sumida et al. 2021).

The Exponomial choice model. Recently, Alptekinoglu and Semple (2016) proposed the Exponomial choice model as a potential alternative to the MNL model. The Exponomial model can be viewed as a specific instantiation of the negative exponential distribution (NED) model, originally introduced by Daganzo (1979, pp. 14–16). Deferring formal definitions to Section 2.1, the Exponomial model states that an arriving customer associates each product with a utility, given by a deterministic component which is perturbed by a (negative) random component. The deterministic component represents the maximum utility that a customer would ever associate with a given product; Alptekinoglu and Semple (2016) appropriately deem this term the "ideal utility". The random component is an i.i.d. exponentially distributed random variable, meant to capture heterogeneity in the customer population. In this setting, arriving customers purchase the product with the largest utility, and when no such product is offered, leave the store without making a purchase. As such, similar to the MNL model, the Exponomial model falls within the class of random utility maximization (RUM) models, whose fundamentals were originally developed by Thurston (1927) to describe whether respondents could differentiate between various psychological stimuli. Interestingly, the Exponomial and MNL choice models share structural similarities, implied by their simple parametric functional form. As explained later on, both have relatively few parameters as well as desirable properties in regards to data-driven estimation.

Alptekinoglu and Semple (2016) were the first to provide operational insights regarding the practicality of the Exponomial model. In particular, they showed that its corresponding estimation problem is efficiently solvable by proving that, with respect to sales data, the log-likelihood function is concave in the model parameters. Furthermore, by investigating a case study based on real household-level panel data set of soup purchases, they found that the Exponomial model outperformed the MNL model in terms of out-of-sample predictive capabilities. In subsequent work, Berbeglia et al. (2018) provided further evidence suggesting that the Exponomial model can be accurately fit to historical sales data. Quite surprisingly, in a series of experiments, the authors discovered that the average out-of-sample predictive performance of the Exponomial model exceeds that of the MNL model. Furthermore, in settings where the training data set is limited or where there is great diversity in customer population preferences, Berbeglia et al. (2018) found that the Exponomial model exhibits the best performance out of all choice models in question.

Research questions. With regards to the assortment optimization problem, Alptekinoglu and Semple (2016) demonstrated that, unlike the MNL model, revenue-ordered assortments are generally sub-optimal for the Exponomial model. They also proposed a greedy backward-elimination heuristic where, starting with an assortment that includes all products, one iteratively eliminates the product whose removal results in the largest increase in expected revenue. Even though this heuristic was shown to perform well on randomly generated instances, no bounds were provided on its deviation from optimality. At present time, it is unknown whether assortment optimization under the Exponomial model is NP-hard, and whether it admits efficient approximation algorithms. This issue could possibly limit the adoption of the Exponomial choice model vis-a-vis the MNL choice model, for which a rich class of assortment problems admit polynomial time algorithms with robust performance guarantees.

Motivated by this state of affairs, we explore two research questions aiming to position the Exponomial choice model as an alternative to the MNL model:

1. *Computational tractability:* Can we devise a provably good approximation algorithm for assortment optimization under the Exponomial model? Complementing this question, are there variants of the latter problem that are provably NP-hard?

2. *Predictive and prescriptive power:* Can we expand the empirical evidence showing that the Exponomial model can be more effective at capturing customer purchasing behavior than the MNL model in certain settings? When should practitioners favor one modeling approach over the other?

1.1. Contributions

In this paper, we first study the tractability of assortment optimization under the Exponomial model, both theoretically and computationally. Next, motivated by our second research question, we present case studies that highlight the predictive and prescriptive power of this model on real-world and synthetic data applications. Below, we provide an overview of our results and findings.

Approximation scheme. Our main technical contribution comes in the form of a fully polynomial-time approximation scheme (FPTAS) for assortment optimization under the Exponomial model. In other words, given an accuracy parameter $\epsilon > 0$, we compute an assortment whose expected revenue is within factor $1-\epsilon$ of optimal, incurring a polynomial running time in the input size and $1/\epsilon$. Our approach is based on a new dynamic programming formulation, where the key insight is that the expected revenue of any assortment can be computed by recursively keeping track of a limited number of basic attributes that "summarize" the subset of products chosen thus far. While rolling this insight into a dynamic programming framework provides a good starting point, the resulting formulation still includes two state variables whose set of potential values is either continuous or exponential in size. Therefore, this initial approach is more of a characterization of optimal assortments than an efficient algorithm. Consequently, at the expense of an ϵ-loss in optimality, we devise a careful discretization of these problematic state variables, which ultimately allows us to identify in polynomial time an assortment whose expected revenue is within $1-\epsilon$ of optimal. The crux of our approach is a novel state pruning method, presented in Section 5, which reduces the bit precision complexity related to the utility specification of the Exponomial choice model. Our FPTAS operates in the general cardinality-constrained setting, noting that our methodology easily extends to incorporate capacity constraints, as explained in Section 8. We also explore the computational efficiency of our method on synthetic instances, demonstrating that our algorithm can be employed to compute near-optimal assortments for instances consisting of up to 30 products within minutes. To keep this paper concise, these numerical results are reported in the online companion, Appendix E.

Computational hardness. Even though we have not been able to prove that the unconstrained assortment optimization problem under the Exponomial model is NP-hard, we still make significant progress towards resolving this open question. To put the latter contributions into context, it is worth mentioning that popular extensions of assortment optimization problems include various operational constraints on the set of products offered. Many common examples in this spirit, such as the cardinality constrained setting described above, can be expressed through Totally Unimodular (TU) constraint matrices, i.e., those where every square submatrix has determinant 0, $+1$, or -1. We refer the reader to Schrijver's book (Schrijver 1998) for an excellent discussion on the theory and practice of TU matrices, as well as to the work of Sumida et al. (2021) for an exhaustive list of relevant assortment features that can be captured by this framework. Our cornerstone intractability result resides in proving that assortment optimization with TU constraints is NP-hard under the Exponomial model. Consequently, we create a strong separation between our computational setting and the MNL-based assortment optimization problem, which was shown by Sumida et al. (2021) to be efficiently solvable under TU constraints. Due to the rather involved nature of this hardness proof, its specifics are provided in the online companion, Appendix A.

Case studies. We evaluate the predictive and prescriptive abilities of the Exponomial model relative to the MNL model on real-world and semi-synthetic data. In this context, the MNL model is a natural benchmark since it is widely used in practice. Additionally, both models are equally parsimonious and share structural similarities, e.g., their choice probabilities are fully specified by a single utility parameter per alternative, which can be estimated from limited data (Ford Jr 1957). Moreover, their maximum likelihood estimator can be expressed as the solution of a convex program. In contrast, more sophisticated alternatives such as the Markov chain, Nested Logit, and mixtures of MNL choice models come with additional computational and statistical hurdles from an estimation standpoint (Şimşek and Topaloglu 2018, Jagabathula et al. 2020).

First, in Section 6, we fit both choice models to data sets describing search queries and bookings of hotel rooms on Expedia's online platform.[1] Comparing their out-of-sample predictions, we find that the Exponomial choice model outperforms the MNL model on various metrics. Next, we examine what characteristics of the choice problem correlate with the performance gap between the Exponomial and MNL choice models. We propose and empirically validate two new metrics, *entropy* and *magnitude*, that are correlated with the gap of predictive performance between these models. We theoretically establish that these metrics are in fact closely related to the notion of *consistent preferences* introduced by Berbeglia et al. (2018). While consistency cannot be directly measured on real-world data, the proposed metrics form easy-to-compute yardsticks for practitioners. Additional case studies on transit and purchase panel data sets are reported in Appendix D.

Second, in Section 7, we measure the expected revenues of optimal assortments prescribed by the Exponomial, MNL, and Markov chain choice models. These models are estimated using semi-synthetic choice data, generated from a real-world survey of preference rankings over sushi items Kamishima (2003). Since the original data set is comprised of individuals who report their full preference list, we can generate counterfactuals for their most preferred item within any given assortment. Hence, we can realistically simulate the expected revenues from different assortment decisions. We find that the Exponomial choice model generates significantly more profitable assortments than the MNL and Markov choice models, with respective revenue gains of 3-4% and 1-2% on average. We provide some evidence that these differences in revenue are due to biases in the predicted choice probabilities.

[1] See url: https://www.kaggle.com/c/expedia-hotel-recommendations.

1.2. Related literature

Assortment optimization problems are fundamental computational tasks in revenue management. The study of efficient algorithms in this context is useful on its own to guide product offering decisions in various applications, such as brick-and-mortar retailing (Kök et al. 2015), e-commerce (Kallus and Udell 2020), and matching (Shi 2015, Gallego et al. 2016). The importance of such algorithmic works has been further elevated in recent years due to their deployment as black-box subroutines in complex stochastic operational problems, including online assortment optimization (Golrezaei et al. 2014), inventory management (Aouad et al. 2018b), and network revenue management problems (Rusmevichientong et al. 2020). Naturally, various computational aspects of the assortment optimization problem depend on the probabilistic assumptions underlying the customer choice behavior.

The MNL model and mixtures thereof. In their seminal work, Talluri and van Ryzin (2004) and Gallego et al. (2004) showed that the optimal assortment under the MNL model takes the form of a revenue-ordered assortment, formed by all products whose revenue is greater than some given threshold. Moreover, Rusmevichientong et al. (2010) proved that the cardinality-constrained version of this problem can be efficiently solved. Later on, Sumida et al. (2021) proposed a linear programming formulation for capturing TU constraints under the MNL model. More generally, there are numerous algorithmic and hardness results related to assortment optimization under a discrete mixture of MNL models. In this context, Bront et al. (2009) proved that assortment optimization is NP-hard and proposed a simple greedy heuristic. Subsequently, Désir et al. (2022) showed that it is NP-hard to approximate the mixture-of-MNL assortment optimization problem within factor $O(n^{1-\epsilon})$, for any fixed $\epsilon > 0$. Despite this strong inapproximability bound, there has been progress in developing polynomial-time algorithms for a few special cases of interest (Désir et al. 2022, Rusmevichientong et al. 2014) as well as upper bounding methods (Feldman and Topaloglu 2015a)

Additional families of choice models. In an attempt to consider general RUM models, one faces strong inapproximability limitations on the resulting assortment optimization problem, as revealed by the $\Omega(n^{1-\epsilon})$ lower bound of Aouad et al. (2018a) for rank-based choice models. Consequently, a vast literature has developed with the objective of expanding the range of RUM choice models for which assortment optimization is solvable in polynomial time, either optimally or near-optimally. For example, the Nested Logit model has received a great deal of attention, beginning with the work of Davis et al. (2014). The NP-hardness status of the resulting assortment optimization problem depends on whether or not customers may leave the store without making a purchase, even after choosing a nest. More recently, Zhang et al. (2020) studied assortment optimization under the paired combinatorial logit (PCL) choice model, where products are grouped into overlapping nests of size at most two, showing in particular that this problem is generally NP-hard. Finally, the computational tractability of the MNL model carries over to the Markov chain model (Blanchet et al. 2016, Feldman and Topaloglu 2015b), where the probabilistic substitution patterns are described by a Markov chain with product-dependent states.

2. The Exponomial Model, Choice Probabilities and Hardness Results

In this section, we initially describe the assortment optimization problem under the Exponomial model, along with additional notation that will be useful later on. We proceed by expressing the choice probabilities in this model through the use of convenient recursive terms, and derive a workable representation for the expected revenue of a given assortment. With these ingredients in place, we highlight some of the unique technical obstacles facing any efficient algorithm for assortment optimization in this context, which are followed by our formal hardness results.

2.1. The Exponomial model

We are given a collection of n products, where the price of product i is denoted by p_i. One of these products represents the no-purchase option, corresponding to the scenario where a customer decides to leave without making a purchase; we refer to the latter as product $\nu \in [n]$, which is associated with a zero price (i.e., $p_\nu = 0$). Each product i is associated with a random utility of $U_i = u_i - Z_i$, where u_i is a fixed ideal utility and $Z_i \sim \exp(\lambda)$ is meant to capture heterogeneity in the customer population. The random variables $Z_1, \ldots, Z_n$ are assumed to be independent.

Any subset of products that contains the no-purchase option ν is called an assortment. For an assortment $S \subseteq [n]$ and a product $i \in S$, we denote by $\pi(i, S)$ the probability that product i is picked when the assortment S is offered. This event occurs when the random utility of product i is greater than the random utility of any other product being offered, namely, $\pi(i, S) = \Pr[U_i = \max_{j \in S} U_j]$.

With respect to this choice model, the expected revenue of a given assortment is $\mathcal{R}(S) = \sum_{i \in S} \pi(i, S) \cdot p_i$. The objective is to identify an assortment whose expected revenue is maximized. For simplicity of presentation, in the remainder of this paper we mainly consider the unconstrained setting, where any subset of products can be offered. In Section 8, we explain how our algorithmic methods can be leveraged to handle cardinality and capacity constraints.

2.2. Definitions and notation

It is easy to verify that the choice probabilities $\pi(i, S)$ remain unchanged when the ideal utilities $u_1, \ldots, u_n$ of all products are uniformly translated. For convenience, we assume without loss of generality that $u_1 \geq \cdots \geq u_n = 0$; note that the no-purchase option ν could have a strictly positive ideal utility u_ν. Furthermore, we may also assume that the rate parameter is $\lambda = 1$, as each utility can be rescaled by λ; this modification preserves the choice probabilities as well.

Assortments via indicators. Rather than representing assortments as subsets, for simplicity of notation we make use of a binary decision variable x_i for each product i, indicating whether this product is offered or not by setting $x_i = 1$ and $x_i = 0$, respectively. It is worth emphasizing that, since the no-purchase option ν is always offered, any assortment sets $x_\nu = 1$. In addition, we use $X(i)$ to denote the number of products offered out of $1, \dots, i$, which is given by $X(i) = \sum_{j=1}^{i} x_j$. Note that, since $u_1 \geq \cdots \geq u_n$, when product i is offered it has the $X(i)$-th largest ideal utility in the assortment.

2.3. Choice probabilities and revenue function

In spite of the seemingly simple structure of the choice probabilities $\pi(i, S) = \Pr[U_i = \max_{j \in S} U_j]$, the latter is known to be expressible via a rather involved expression, derived by Daganzo (1979) as well as by Alptekinoglu and Semple (2016). As previously mentioned, it is notationally convenient for our purposes to index the set of products such that $u_1 \geq \cdots \geq u_n$, which is precisely the reverse order of the indexing notation utilized by the above-mentioned authors. In this case, letting $G_i(x) = \frac{\exp(-\sum_{j=1}^{i}(u_j - u_i) \cdot x_j)}{X(i)}$, with the convention that $G_i(x) = 0$ whenever $X(i) = 0$, the probability $\pi(i, x)$ that product i is picked out of the assortment x is given by

$$\pi(i, x) = \left(G_i(x) - \sum_{j=i+1}^{n} \frac{G_j(x) \cdot x_j}{X(j-1)} \right) \cdot x_i \ .$$

In turn, we obtain the following expected revenue function

$$\begin{aligned} \mathcal{R}(x) &= \sum_{i=1}^{n} \pi(i, x) \cdot p_i \\ &= \sum_{i=1}^{n} \left(G_i(x) - \sum_{j=i+1}^{n} \frac{G_j(x) \cdot x_j}{X(j-1)} \right) \cdot p_i x_i \\ &= \sum_{i=1}^{n} G_i(x) \cdot \left(p_i - \frac{1}{X(i-1)} \cdot \sum_{j=1}^{i-1} p_j x_j \right) \cdot x_i \ . \end{aligned} \tag{1}$$

We refer to the inner term in the last summation, $G_i(x) \cdot (p_i - \frac{1}{X(i-1)} \cdot \sum_{j=1}^{i-1} p_j x_j) \cdot x_i$, as the contribution of product i to the expected revenue $\mathcal{R}(x)$. It is important to emphasize that, very much in contrast to common intuition, since we may have $p_i < \frac{1}{X(i-1)} \cdot \sum_{j=1}^{i-1} p_j x_j$, the contribution of a given product to $\mathcal{R}(x)$ can actually be negative, even in an optimal assortment.

2.4. Technical obstacle: Bit precision and pseudo-polynomial time

To highlight a hidden technical difficulty that should be surpassed by any polynomial-time algorithm, it is worth focusing on a toy example, that appears to be asking a trivial question. For this purpose, suppose we are given the following instance:

- The set of products is $\{1, 2, 3\}$, of which $\nu = 3$ is the no-purchase option.

- The prices of these products are $p_1 > p_2 > p_3 = 0$.
- The ideal utilities are $u_1 > u_2 > u_3 = 0$.

Let us consider the problem of identifying the optimal assortment in this seemingly simple case. Since the no-purchase option $\nu = 3$ is always offered, there are only 4 feasible assortments: $\{3\}$, $\{1,3\}$, $\{2,3\}$, and $\{1,2,3\}$. Out of these options, the first clearly has zero revenue, whereas each of the other three is associated with an expected revenue that can be obtained by representation (1), resulting in:

$$\begin{aligned}
\mathcal{R}(\{1,3\}) &= \left(1 - \frac{e^{-u_1}}{2}\right) \cdot p_1 \ , \\
\mathcal{R}(\{2,3\}) &= \left(1 - \frac{e^{-u_2}}{2}\right) \cdot p_2 \ , \\
\mathcal{R}(\{1,2,3\}) &= p_1 - \frac{e^{-(u_1-u_2)}}{2} \cdot (p_1 - p_2) - \frac{e^{-(u_1+u_2)}}{3} \cdot \frac{p_1 + p_2}{2} \ .
\end{aligned}$$

Therefore, the optimal assortment can easily be identified by computing the above quantities and comparing between them. This idea is clearly correct, but unfortunately, its running time is not polynomial in the input size.

To understand the latter statement, first note that the terms e^{-u_1}, e^{-u_2}, $e^{-(u_1-u_2)}$, and $e^{-(u_1+u_2)}$ do not necessarily correspond to rational numbers, even for rational ideal utilities. This observation by itself would require bit precision arguments in order to determine the maximum revenue out of $\mathcal{R}(\{1,3\})$, $\mathcal{R}(\{2,3\})$, and $\mathcal{R}(\{1,2,3\})$. However, even more worryingly, suppose that instead of e^{-u_1}, $e^{-u_2}, \ldots$ we would have had 2^{-u_1}, $2^{-u_2}, \ldots$, with all ideal utilities being integers. Merely specifying a quantity such as 2^{-u_1} requires u_1 many bits, which is generally exponential in the input size, given in our toy example by $\log u_1 + \log u_2 + \log p_1 + \log p_2 + O(1)$. Based on the previous example, we have just learned that polynomial-time approximation algorithms for assortment optimization under the Exponomial choice model cannot directly perform even the simplest operations related to revenue evaluation. As further explained in Sections 3-5, these technical subtleties will only make up part of the difficulties that our FPTAS will have to overcome.

2.5. Hardness results

In spite of our best efforts, we have not been able to prove that the unconstrained assortment optimization problem under the Exponomial model is NP-hard. We currently believe that determining the exact complexity status of this problem is a challenging open question. Nevertheless, in what follows, we make significant progress towards answering this question, by presenting a number of mechanisms and gadgets for establishing intractability results when facing additional constraints on the set of products offered.

Capacity constraints. We first consider the more manageable objective of proving NP-hardness for the capacity-constrained assortment optimization setting. Here, we are no longer allowed to offer any subset of products. Instead, each product i is associated with a non-negative size s_i, and we are given an upper bound B on the total size of the selected assortment. The objective is to compute a feasible assortment whose expected revenue is maximized. In Appendix A.1, we prove the next theorem.

THEOREM 1. *Capacity-constrained assortment optimization under the Exponomial choice model is NP-hard.*

Totally unimodular constraints. We now turn our attention to a significantly more involved reduction, allowing us to derive NP-hardness for totally unimodular (TU) constraints. The latter form a general framework for capturing a wide range of practical constraints such as cardinality, precedence, display locations, and quality consistent pricing (see, for example, Sumida et al. (2021)). For convenience of notation, let $x^S \in \{0,1\}^n$ be the incidence vector of the assortment S, where $x_i^S = 1$ if $i \in S$ and $x_i^S = 0$ otherwise. In this setting, given a TU matrix A and an integer vector b, we are allowed to offer only assortments S that satisfy $Ax^S \leq b$. Once again, the objective is to compute a feasible assortment whose expected revenue is maximized. In Appendix A.2, we establish the following hardness result.

THEOREM 2. *TU-constrained assortment optimization under the Exponomial choice model is NP-hard.*

Theorem 2 implies a strong separation between the MNL-based assortment optimization problem and the Exponomial-based setting. Despite this negative result, as explained in Section 8, our dynamic programming approach can be leveraged to attain an FPTAS under a cardinality constraint, which falls under the framework of TU-constrained assortment problems. Using similar ideas, our approach can be generalized to obtain an FPTAS for a general capacity constraint. That said, similar to the unconstrained setting, we believe that determining the exact complexity status of having a cardinality constraint is challenging and present this open question in Section 8.

3. The Continuous Dynamic Program

Based on the foundations built in Section 2, we proceed by explaining how to formulate the assortment optimization problem as a suitable dynamic program with a continuum of states. This characterization will serve as a starting point for developing our discretization method in Sections 4 and 5, eventually leading to a fully polynomial-time approximation scheme.

3.1. Intuition and states

Preliminary observations. Motivated by the revenue representation (1), let us consider the ingredients needed in order to evaluate the contribution of product i to $\mathcal{R}(x)$:

- First, to calculate $G_i(x)$, we need to know $X(i) = X(i-1) + x_i$, which stands for the number of products offered out of $1, \dots, i$. In addition, the value $\sum_{j=1}^{i}(u_j - u_i) \cdot x_j = \sum_{j=1}^{i-1}(u_j - u_{i-1}) \cdot x_j + X(i-1) \cdot (u_{i-1} - u_i)$ is also required; this is the sum of differences between the utilities of all products offered out of $1, \dots, i$ and that of product i.
- Furthermore, we need to know $\frac{1}{X(i-1)} \cdot \sum_{i=j}^{i-1} p_j x_j$, which is the average price over all products offered out of $1, \dots, i-1$.

The important observation is that each of these ingredients can be thought of as a compactly representable function of our decisions for products $1, \dots, i-1$, along with the individual decision of whether or not to pick product i.

States. Consequently, each state of our dynamic program is described by the following four parameters:

1. Index of the current product, i, taking values in $\{1, \dots, n\}$.
2. The size N of the assortment picked thus far, out of $1, \dots, i-1$. This quantity corresponds to $X(i-1)$ and takes values in $\{0, \dots, n-1\}$.
3. The sum of differences U^{Σ} between the utilities of all products picked out of $1, \dots, i-1$ and that of $i-1$, corresponding to $\sum_{j=1}^{i-1}(u_j - u_{i-1}) \cdot x_j$. This parameter is clearly bounded within the interval $[0, nu_1]$.
4. The average price P^{avg} over the products picked out of $1, \dots, i-1$. Letting $p_{\max}$ designate the maximal price of any product, P^{avg} is bounded within $[0, p_{\max}]$.

It is worth pointing out that the continuous dynamic program we are about to propose is not algorithmic in nature, due to allowing both U^{Σ} and P^{avg} to take a continuum of values. The actual set of values for these parameters is finite, as each subset of products corresponds to a single value. However, simple examples demonstrate that the number of such values could very well be $\Omega(2^n)$. As such, our main objective is to devise a dynamic program that characterizes optimal assortments and serves as a starting point to obtain an FPTAS via an appropriate discretization of the state space.

3.2. Recursive equations

From this point on, we use $\mathcal{C}(i, N, U^{\Sigma}, P^{\text{avg}})$ to denote the maximum sum of revenue contributions that can be obtained from products $i, \dots, n$, given that N products were picked out of $1, \dots, i-1$, with a sum of utility differences U^{Σ} and an average price of P^{avg}. Based on this definition, the optimal assortment clearly corresponds to $\mathcal{C}(1, 0, 0, 0)$. In order to obtain the recursive equations for our dynamic program, we first define $G(i,N,U^{\Sigma}) = \frac{\exp(-U^{\Sigma}-N\cdot(u_{i-1}-u_i))}{N+1}$ for $i \geq 2$, with the convention that $G(1,0,0)=1$. With this notation, $\mathcal{C}(i,N,U^{\Sigma},P^{\text{avg}})$ can be computed via backward recursion as follows.

Terminal states (with $i=n$). In this case, we can either pick product n and obtain a revenue contribution of $G(n,N,U^{\Sigma})\cdot(p_n - P^{\text{avg}})$, or leave out this product, leading to zero revenue contribution. Therefore,

$$\mathcal{C}\left(n,N,U^{\Sigma},P^{\text{avg}}\right) = \max\left\{ \underbrace{G\left(n,N,U^{\Sigma}\right)\cdot(p_n - P^{\text{avg}})}_{\text{pick } n}, \underbrace{0}_{\text{do not pick } n} \right\}. \tag{2}$$

Non-terminal states (with $1 \leq i \leq n-1$). In this case, there are two options. First, we can pick product i, obtain an immediate revenue contribution of $G(i,N,U^{\Sigma})\cdot(p_i - P^{\text{avg}})$, and guarantee that the future sum of contributions is $\mathcal{C}(i+1,N+1,U^{\Sigma}+N\cdot(u_{i-1}-u_i),\frac{NP^{\text{avg}}+p_i}{N+1})$. Alternatively, leaving out product i results in zero immediate contribution, ensuring that the future sum of contributions is $\mathcal{C}(i+1,N,U^{\Sigma}+N\cdot(u_{i-1}-u_i),P^{\text{avg}})$. Consequently,

$$\begin{aligned}
&\mathcal{C}\left(i,N,U^{\Sigma},P^{\text{avg}}\right) \\
&\quad = \max\left\{ \underbrace{G\left(i,N,U^{\Sigma}\right)\cdot(p_i - P^{\text{avg}}) + \mathcal{C}\left(i+1,N+1,U^{\Sigma}+N\cdot(u_{i-1}-u_i),\frac{NP^{\text{avg}}+p_i}{N+1}\right)}_{\text{pick } i}, \right. \\
&\qquad\qquad \left. \underbrace{\mathcal{C}\left(i+1,N,U^{\Sigma}+N\cdot(u_{i-1}-u_i),P^{\text{avg}}\right)}_{\text{do not pick } i} \right\}.
\end{aligned} \tag{3}$$

3.3. Structural differences with the MNL model

To gain intuition on equation (3), we notice that our dynamic programming formulation exploits various properties of the Exponomial model, which are closely related to its substitution patterns. Specifically, our dynamic program utilizes the average price P^{avg} and the sum of utility differences U^{Σ} over previously stocked products as state variables, among others. This state description is directly related to the market share cannibalization effects described by the Exponomial choice model:

1. The substitution in market shares exerted by any given product on higher utility products is *uniform*. This property is reflected in equation (3), where the terms that negatively contribute to the immediate reward correspond to the cannibalization of market shares of higher utility products. Since the "amount of cannibalization" is uniform, the prices of previously stocked products are uniformly averaged.

2. Moreover, this substitution is a function of the product rank and the total utility gap between the product in question and all higher utility products stocked. In particular, *this dependency does not involve lower utility products subsequently stocked in the assortment.*

There are striking differences between these properties and the substitution effects captured by the MNL model. First, by the Independence of Irrelevant Alternatives property, the reduction in market shares due to stocking an additional product is proportional to the MNL weights rather than uniform, in sharp contrast with property 1. Indeed, the amount of cannibalization depends on the sum of MNL weights, which forms a natural counterpart to the total utility gap of property 2. However, the product's rank does not have any direct effect on the amount of substitution under the MNL model.

4. The Approximate Dynamic Program

In this section, we convert our continuous dynamic program into a more tractable counterpart, which only requires polynomially many discrete states. Naturally, this approximate formulation results in a small loss of optimality. However, noting that our method eventually guarantees an FPTAS, there is an explicit tradeoff between its performance guarantee and running time, controlled by an error parameter $\epsilon > 0$.

Outline. Our current goal is to discretize the continuous state variables P^{avg} and U^{Σ} appearing in the dynamic program (3), while restricting attention to polynomially many distinct values for each parameter. This approach proceeds in two steps.

1. *Discretization of average prices (Section 4.1):* First, we discretize the parameter P^{avg} that stands for the average price over the products picked thus far. Given an error parameter $\epsilon > 0$, we explain how to construct a set $\mathcal{P}_\epsilon$ of polynomially many representative values for the average price. This set is then utilized to create an approximate version $\mathcal{C}_{\mathcal{P}_\epsilon}$ of the continuous dynamic program $\mathcal{C}$ where, rather than allowing P^{avg} to take arbitrary values in $[0, p_{\max}]$, we "round" the average price in each step to a nearby value in $\mathcal{P}_\epsilon$. We show that an optimal assortment for $\mathcal{C}_{\mathcal{P}_\epsilon}$ provides a $(1-\epsilon)$-approximation with respect to the original program $\mathcal{C}$, where P^{avg} is unrestricted.

2. *Discretization of sum-of-utility-differences (Section 4.2):* Next, we move on to discretize the second continuous parameter, U^{Σ}, which represents the sum of differences between the utilities of all products picked so far and that of the current product. Given an error parameter $\epsilon > 0$, we explain how to slightly modify the ideal utilities u_i, such that the sum-of-utility-differences parameter U^{Σ} is restricted to an appropriately constructed set $\mathcal{U}_\epsilon$. As in Step 1, this discrete set is utilized to create a further approximate version $\mathcal{C}_{\mathcal{P}_\epsilon, \mathcal{U}_\epsilon}$ of the original dynamic program, where U^{Σ} is no longer allowed to take arbitrary values in $[0, nu_1]$. Our analysis proves that an optimal assortment for $\mathcal{C}_{\mathcal{P}_\epsilon, \mathcal{U}_\epsilon}$ provides a $(1-\epsilon)$-approximation with respect to the original program $\mathcal{C}$.

In spite of advancing toward a computationally tractable dynamic program, one remaining challenge would be that the set $\mathcal{U}_\epsilon$ is not polynomially sized. Indeed, in Step 2, we still need a *pseudo-polynomial* number of representative values in $\mathcal{U}_\epsilon$ to guarantee that the approximate dynamic program $\mathcal{C}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}$ loses only an $O(\epsilon)$-factor in optimality. Another related challenge is that, at first glance, our recursive equation cannot be efficiently solved due to the bit precision complexity mentioned in Section 2.4. Hence, the final piece of our algorithm is given in Section 5, where we show how to bypass these challenges using suitable state pruning methods.

4.1. Discretization of average prices

Defining $\mathcal{P}_\epsilon$. As previously mentioned, for any assortment x, the average price $\frac{1}{X(i-1)} \cdot \sum_{j=1}^{i-1} p_j x_j$ resides within the interval $[0, p_{\max}]$. However, when this quantity is strictly greater than 0, it is actually lower bounded by $\frac{p_{\min}}{n}$, where $p_{\min}$ is the minimum non-zero price. Given an error parameter $\epsilon > 0$, starting with $\mathcal{P}_\epsilon = \{0\}$, we add to this set all powers of $1+\theta$ in the interval $[\frac{p_{\min}}{n}, p_{\max}]$, where $\theta = \frac{\epsilon}{2n^3}$. As a result, the cardinality of $\mathcal{P}_\epsilon$ is indeed polynomial in the input size, since

$$|\mathcal{P}_\epsilon| = O\left(\log_{1+\theta}\left(\frac{np_{\max}}{p_{\min}}\right)\right) = O\left(\frac{n^3}{\epsilon} \cdot \log\left(\frac{np_{\max}}{p_{\min}}\right)\right) .$$

Defining $\mathcal{C}_{\mathcal{P}_\epsilon}$. In order to derive an approximate version $\mathcal{C}_{\mathcal{P}_\epsilon}$ of the dynamic program $\mathcal{C}$, rather than using its original state space $(i, N, U^\Sigma, P^{\text{avg}})$, we impose that the average price parameter P^{avg} takes values only in $\mathcal{P}_\epsilon$. Therefore, the latter parameter will no longer correspond to the average price of a certain subset of products, but instead should be thought of as a sufficiently accurate estimate, which is why we designate it by $\tilde{P}^{\text{avg}}$.

Suppose that our approximate dynamic program $\mathcal{C}_{\mathcal{P}_\epsilon}$ arrives at step i to the state $(i, N, U^\Sigma, \tilde{P}^{\text{avg}})$, with $\tilde{P}^{\text{avg}} \in \mathcal{P}_\epsilon$. Then, if we decide to pick product i, the original recursive equations (3) would use $\frac{N\tilde{P}^{\text{avg}}+p_i}{N+1}$ as the updated average, which is indeed feasible for $\mathcal{C}$, due to allowing the average price to take any value in $[0, p_{\max}]$. However, an update of the form $\frac{N\tilde{P}^{\text{avg}}+p_i}{N+1}$ is generally infeasible for $\mathcal{C}_{\mathcal{P}_\epsilon}$, since $\mathcal{P}_\epsilon$ is not closed under such operations. Our method to get around this obstacle is to resume with the updated estimate $\mathcal{P}_\epsilon(\frac{N\tilde{P}^{\text{avg}}+p_i}{N+1})$, where $\mathcal{P}_\epsilon(\cdot)$ is an operator that rounds its argument up to the nearest value in $\mathcal{P}_\epsilon$. Therefore, the general case recursive equation becomes

$$\begin{aligned}\mathcal{C}_{\mathcal{P}_\epsilon}\left(i, N, U^\Sigma, \tilde{P}^{\text{avg}}\right) \quad = \quad \max\Bigg\{ & \mathcal{C}_{\mathcal{P}_\epsilon}\left(i+1, N, U^\Sigma + N\cdot(u_{i-1}-u_i), \tilde{P}^{\text{avg}}\right), \\ & G\left(i, N, U^\Sigma\right)\cdot\left(p_i - \tilde{P}^{\text{avg}}\right) + \mathcal{C}_{\mathcal{P}_\epsilon}\left(i+1, N+1, U^\Sigma + N\cdot(u_{i-1}-u_i), \mathcal{P}_\epsilon\left(\frac{N\tilde{P}^{\text{avg}}+p_i}{N+1}\right)\right)\Bigg\} . \end{aligned} \tag{4}$$

For the terminal case of $i = n$, the recursive equation is identical to (2), i.e.,

$$\mathcal{C}_{\mathcal{P}_\epsilon}\left(n, N, U^\Sigma, \tilde{P}^{\text{avg}}\right) = \max\left\{G\left(n, N, U^\Sigma\right)\cdot\left(p_n - \tilde{P}^{\text{avg}}\right), 0\right\} . \tag{5}$$

Approximation guarantee. The remainder of this section is devoted to showing that $\mathcal{C}_{\mathcal{P}_\epsilon}$ provides a $(1-\epsilon)$-approximation for $\mathcal{C}$. To formalize this notion, we first provide explicit expressions for the sequence of states traversed by following the decisions of any assortment with respect to these dynamic programs.

For the original dynamic program $\mathcal{C}$ and for an assortment x, we use $\mathcal{C}(x)$ to denote the objective value attained by x with respect to $\mathcal{C}$, where x_i indicates whether product i is picked in step i or not. It is easy to verify that, by following these decisions, equations (2) and (3) imply that we reach step i at state $(i, X(i-1), \sum_{j=1}^{i-1}(u_j - u_{i-1}) \cdot x_j, \frac{1}{X(i-1)} \cdot \sum_{i=j}^{i-1} p_j x_j)$, and therefore,

$$\mathcal{C}(x) = \sum_{i=1}^{n} G_i(x) \cdot \left(p_i - \frac{1}{X(i-1)} \cdot \sum_{j=1}^{i-1} p_j x_j \right) \cdot x_i = \mathcal{R}(x) \ . \tag{6}$$

Similarly, let $\mathcal{C}_{\mathcal{P}_\epsilon}(x)$ be the objective value attained by x with respect to the approximate dynamic program $\mathcal{C}_{\mathcal{P}_\epsilon}$. In this case, by following the decisions specified by x, equations (4) and (5) show that we reach step i at state $(i, X(i-1), \sum_{j=1}^{i-1}(u_j - u_{i-1}) \cdot x_j, \mathrm{avg}_{i-1}(x))$. Here, $\mathrm{avg}_{i-1}(x)$ is recursively defined by $\mathrm{avg}_0(x) = 0$ and

$$\mathrm{avg}_{i-1}(x) = \mathcal{P}_\epsilon \left(\frac{X(i-2) \cdot \mathrm{avg}_{i-2}(x) + p_{i-1} x_{i-1}}{X(i-1)} \right)$$

for $i = 2, \ldots, n$. Consequently, the objective value attained by x is precisely

$$\mathcal{C}_{\mathcal{P}_\epsilon}(x) = \sum_{i=1}^{n} G_i(x) \cdot \left(p_i - \mathrm{avg}_{i-1}(x) \right) \cdot x_i \ . \tag{7}$$

The next claim, whose proof is given in Appendix B.1, relates our estimate $\mathrm{avg}_{i-1}(x)$ to the actual average price $\frac{1}{X(i-1)} \cdot \sum_{i=j}^{i-1} p_j x_j$.

Lemma 1. *For any assortment x, and for every $i \in [n]$,*

$$\frac{1}{X(i-1)} \cdot \sum_{j=1}^{i-1} p_j x_j \leq \mathrm{avg}_{i-1}(x) \leq \frac{(1+\theta)^{i-1}}{X(i-1)} \cdot \sum_{j=1}^{i-1} p_j x_j \ .$$

Building on this property, the next theorem shows that $\mathcal{C}_{\mathcal{P}_\epsilon}$ provides a $(1-\epsilon)$-approximation for $\mathcal{C}$; the proof is detailed in Appendix B.2.

Theorem 3. *Let $\tilde{x}$ be an optimal assortment for $\mathcal{C}_{\mathcal{P}_\epsilon}$, and let x^* be an optimal assortment for the original objective function $\mathcal{R}$. Then, $\mathcal{R}(\tilde{x}) \geq (1-\epsilon) \cdot \mathcal{R}(x^*)$.*

4.2. Discretization of sum-of-utility-differences

We remind the reader that, for any assortment x, the sum of utility differences $\sum_{j=1}^{i}(u_j - u_i) \cdot x_j$ resides within the interval $[0, nu_1]$. Given an error parameter $\epsilon > 0$ and letting $\delta = \frac{\epsilon}{2n}$, we define the modified utility of product i as $\tilde{u}_i = \lfloor u_i \rfloor_\delta$, where $\lfloor \cdot \rfloor_\delta$ is an operator that rounds its argument down to the nearest integer multiple of δ. Accordingly, we construct the set $\mathcal{U}_\epsilon$ by taking all integer multiples of δ within $[0, nu_1]$; hence, for any subset of products, their sum of modified utility differences resides in $\mathcal{U}_\epsilon$. Still, it is worth pointing out that the cardinality of $\mathcal{U}_\epsilon$ is pseudo-polynomial in the input size, since $|\mathcal{U}_\epsilon| = O(\frac{n^2 u_1}{\epsilon})$.

Defining $\mathcal{C}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}$. We now propose an additional layer of approximation on top of the dynamic program $\mathcal{C}_{\mathcal{P}_\epsilon}$, resulting in a finite-state program to which we refer as $\mathcal{C}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}$. Here, rather than having states of the form $(n, N, U^\Sigma, \tilde{P}^{\text{avg}})$, where the sum-of-utility-differences parameter U^Σ is allowed to take any value in $[0, nu_1]$, we replace each utility u_i by its modified version $\tilde{u}_i$, which forces U^Σ to take values only in $\mathcal{U}_\epsilon$. Similarly to its counterpart $\tilde{P}^{\text{avg}}$, this parameter no longer corresponds to an actual sum of utility differences of a certain subsets of products, and is meant to serve as a sufficiently accurate estimate, designated by $\tilde{U}^\Sigma$. As a result, the recursive equations are nearly identical to those of $\mathcal{C}_{\mathcal{P}_\epsilon}$, except for plugging-in $\tilde{u}_i$ instead of u_i. In other words, the general case equation becomes

$$\begin{aligned} \mathcal{C}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}\left(i, N, \tilde{U}^\Sigma, \tilde{P}^{\text{avg}}\right) \quad = \quad \max\Bigg\{ & \mathcal{C}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}\left(i+1, N, \tilde{U}^\Sigma + N\cdot(\tilde{u}_{i-1} - \tilde{u}_i), \tilde{P}^{\text{avg}}\right), \\ & \tilde{G}\left(i, N, \tilde{U}^\Sigma\right)\cdot\left(p_i - \tilde{P}^{\text{avg}}\right) + \mathcal{C}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}\left(i+1, N+1, \tilde{U}^\Sigma + N\cdot(\tilde{u}_{i-1} - \tilde{u}_i), \mathcal{P}_\epsilon\left(\frac{N\tilde{P}^{\text{avg}} + p_i}{N+1}\right)\right)\Bigg\} \end{aligned} \tag{8}$$

where $\tilde{G}(i, N, \tilde{U}^\Sigma) = \frac{\exp(-\tilde{U}^\Sigma - N\cdot(\tilde{u}_{i-1} - \tilde{u}_i))}{N+1}$. For the terminal case of $i = n$, the recursive equation is

$$\mathcal{C}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}\left(n, N, \tilde{U}^\Sigma, \tilde{P}^{\text{avg}}\right) = \max\left\{\tilde{G}\left(n, N, \tilde{U}^\Sigma\right)\cdot\left(p_n - \tilde{P}^{\text{avg}}\right), 0\right\} . \tag{9}$$

Approximation guarantee. To establish the performance guarantee attained by optimal assortments for $\mathcal{C}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}$, we begin by presenting an explicit expression for the sequence of states traversed by following the decisions of any assortment with respect to $\mathcal{C}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}$. It is worth noting that the corresponding sequence of states with respect to $\mathcal{C}_{\mathcal{P}_\epsilon}$ has already been presented in Section 4.1.

For an assortment x, we use $\mathcal{C}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}(x)$ to denote the objective value attained by x with respect to $\mathcal{C}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}$, where x_i indicates whether product i is picked in step i or not. It is easy to verify that, by following these decisions, equations (8) and (9) imply that we reach step i at state $(i, X(i-1), \sum_{j=1}^{i-1}(\tilde{u}_j - \tilde{u}_{i-1})\cdot x_j, \text{avg}_{i-1}(x))$. Consequently, the objective value attained by x is

$$\mathcal{C}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}(x) = \sum_{i=1}^{n} \tilde{G}_i(x)\cdot\left(p_i - \text{avg}_{i-1}(x)\right)\cdot x_i \ . \tag{10}$$

Unlike the objective values for $\mathcal{C}$ and $\mathcal{C}_{\mathcal{P}_\epsilon}$ (see (6) and (7), respectively), here we have a coefficient of $\tilde{G}_i(x) = \frac{\exp(-\sum_{j=1}^{i}(\tilde{u}_j - \tilde{u}_i)\cdot x_j)}{X(i)}$ rather than $G_i(x) = \frac{\exp(-\sum_{j=1}^{i}(u_j - u_i)\cdot x_j)}{X(i)}$, due to the modified utilities. The next claim, whose proof is given in Appendix B.4, relates our estimate $\sum_{j=1}^{i}(\tilde{u}_j - \tilde{u}_i)\cdot x_j$ to the actual sum of utilities $\sum_{j=1}^{i}(u_j - u_i)\cdot x_j$.

Lemma 2. *For any assortment x, and for every $i \in [n]$,*

$$\sum_{j=1}^{i}(u_j - u_i)\cdot x_j - i\delta \leq \sum_{j=1}^{i}(\tilde{u}_j - \tilde{u}_i)\cdot x_j \leq \sum_{j=1}^{i}(u_j - u_i)\cdot x_j + i\delta \ .$$

We are now ready to establish that $\mathcal{C}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}$ provides a $(1-\epsilon)^3$-approximation for $\mathcal{C}$. Specifically, we conclude our analysis by showing that, when an optimal assortment for $\mathcal{C}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}$ is evaluated as a candidate solution to the original assortment optimization problem, its expected revenue is within factor $(1-\epsilon)^3$ of optimal. The proof of this claim appears in Appendix B.5.

Theorem 4. *Let $\bar{x}$ be an optimal assortment for $\mathcal{C}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}$, and let x^* be an optimal assortment for the original objective function $\mathcal{R}$. Then, $\mathcal{R}(\bar{x}) \geq (1-\epsilon)^3 \cdot \mathcal{R}(x^*)$.*

5. Devising an FPTAS via State Pruning

In this section, we provide the final piece of the puzzle by forcing the finite-state dynamic program $\mathcal{C}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}$ to attain a truly polynomial running time. To this end, we develop a state pruning method by combining a new initialization and termination rule for our dynamic program. These ideas may be of broader interest in converting pseudo-polynomial to polynomial time reductions in assortment optimization problems.

5.1. Intermediate summary

At the moment, the approximate program $\mathcal{C}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}$ is a well-defined approximation scheme, but unfortunately, not a polynomial-time one. A close inspection of our construction reveals two inherent reasons for the running time of our current algorithm to be pseudo-polynomial:

1. *Number of states.* Each state of the dynamic program $\mathcal{C}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}$ consists of four parameters, $(i, N, \tilde{U}^\Sigma, \tilde{P}^{\text{avg}})$. In Section 3, we have noticed that the current product index i and the size N of the assortment picked thus far take $O(n)$ values each, whereas Section 4.1 allows our average price estimate $\tilde{P}^{\text{avg}}$ to take values in the set $\mathcal{P}_\epsilon$, of cardinality $O(\frac{n^3}{\epsilon}\cdot\log(\frac{np_{\max}}{p_{\min}}))$. However, Section 4.2 lets the sum-of-utility-differences estimate $\tilde{U}^\Sigma$ take values in $\mathcal{U}_\epsilon$, of cardinality $O(\frac{n^2 u_1}{\epsilon})$. Therefore, the overall number of states is polynomial in the maximum utility u_1, rather than in the input size.

2. *$\tilde{G}$-coefficients.* The recursive equations (8) and (9) for computing the function value $\mathcal{C}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}(i, N, \tilde{U}^\Sigma, \tilde{P}^{\text{avg}})$ involve the term $\tilde{G}(i, N, \tilde{U}^\Sigma) = \frac{\exp(-\tilde{U}^\Sigma - N\cdot(\tilde{u}_{i-1}-\tilde{u}_i))}{N+1}$. Thus, since the sum-of-utility-differences estimate $\tilde{U}^\Sigma$ could be as large as nu_1, merely specifying the term $\tilde{G}(i, N, \tilde{U}^\Sigma)$ for some states requires $\Omega(nu_1)$-many bits, which is again polynomial in u_1.

The high-level idea. To arrive at a true FPTAS, our approach is based on showing that most of the possible values in $\mathcal{U}_\epsilon$ are "irrelevant" for the parameter $\tilde{U}^\Sigma$. In essence, when the approximate dynamic program $\mathcal{C}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}$ leaves a certain set of polynomially many states, we prove that a $(1-\epsilon)$-fraction of the optimal objective value has already been secured. The remaining states traversed from that point on are precisely those that lead to a pseudo-polynomial running time, and can be avoided by starting $\mathcal{C}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}$ at a carefully picked initial state (instead of $(1,0,0,0)$), coupled with an appropriate termination rule. Furthermore, due to these enhancements, we are able to argue that the resulting dynamic program makes use of $\tilde{G}$-coefficients that are all within the same order of magnitude. Consequently, by factoring out a certain exponent (rather than explicitly computing it), we are left with scaled $\tilde{G}$-coefficients that can be specified using polynomially many bits.

5.2. State pruning method

Single-product assortments. For any product $i \in [n]$, let $x_{[i]}$ be the assortment that stocks only product i and the no-purchase option ν. Recalling that $\tilde{G}_i(x) = \frac{\exp(-\sum_{j=1}^{i}(\tilde{u}_j - \tilde{u}_i)\cdot x_j)}{X(i)}$, it follows that

$$\tilde{G}_i(x_{[i]}) = \begin{cases} 1, & \text{if } i \leq \nu \\ \frac{1}{2}e^{-(\tilde{u}_\nu - \tilde{u}_i)} & \text{if } i > \nu \end{cases}$$

Hence, since $\tilde{u}_\nu$ and $\tilde{u}_i$ are both multiples of δ, this quantity can be written as $\tilde{G}_i(x_{[i]}) = \frac{1}{d_i} \cdot e^{-k_i\delta}$, where $d_i \in \{1,2\}$ and k_i is an integer between 0 and $\frac{u_1}{\delta}$. An immediate implication of this observation is that we can compare any pair of expressions $\tilde{G}_{i_1}(x_{[i_1]}) \cdot p_{i_1}$ and $\tilde{G}_{i_2}(x_{[i_2]}) \cdot p_{i_2}$ without explicitly evaluating these terms. Instead, it suffices to compare the corresponding exponents $-k_{i_1}\delta + \ln(\frac{p_{i_1}}{d_{i_1}})$ and $-k_{i_2}\delta + \ln(\frac{p_{i_2}}{d_{i_2}})$; this can easily be done in polynomial time. Therefore, our first step consists of identifying in polynomial time the product $\ell \in [n]$ that maximizes $\tilde{G}_\ell(x_{[\ell]}) \cdot p_\ell$. We note in passing that the latter quantity is generally different from the expected revenue $\mathcal{R}(x_{[\ell]})$, meaning that product ℓ is not necessarily the one corresponding to the best single-product assortment.

New initial state. Up until now, the dynamic program $\mathcal{C}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}$ starts at the initial state $(1,0,0,0)$. Namely, the first product to be considered is 1, meaning that no products were picked thus far, leading to zero values for the sum-of-utility-differences and average price parameters. Letting $\bar{x}$ be a fixed optimal assortment for $\mathcal{C}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}$, we guess the minimum-index product $f \neq \nu$ that is picked by this assortment (i.e., $\bar{x}_f = 1$). Technically speaking, "guessing" means trying all possible values that could ever be taken by f (i.e., all $n-1$ products different from ν), running the algorithm from this point on with the current guess, and finally, picking the most profitable assortment computed. Clearly, the performance guarantee obtained would be at least as good as the one where the correct guess (namely, the unknown product f) is tested, since it is one of the values considered. To define our new initial state, we embed this guess into the dynamic program, starting at either $\mathcal{S}_f = (f,0,0,0)$ when $f < \nu$, or at $\mathcal{S}_f = (f,1,\tilde{u}_\nu - \tilde{u}_{f-1},0)$ when $f > \nu$. We also add a special recursive equation for this state, that forces any assortment to pick product f. To avoid confusion, we denote the resulting dynamic program by $\mathcal{C}^{\mathcal{S}_f}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}$.

New terminating condition. In order to ensure that our dynamic program runs in true polynomial time, we introduce one final adjustment. Starting from the initial state $\mathcal{S}_f$, rather than formulating the dynamic program $\mathcal{C}^{\mathcal{S}_f}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}$ over all possible states $(i, N, \tilde{U}^\Sigma, \tilde{P}^{\text{avg}})$, we restrict attention to states with $\tilde{G}(i, N, \tilde{U}^\Sigma) \geq e^{-k_\ell\delta-\Delta}$, which are referred to as being profitable. Here, $\Delta = \ln(\frac{4n}{\epsilon} \cdot \frac{p_{\max}}{p_{\min}})$, where $p_{\max}$ and $p_{\min}$ are the maximal price and minimum non-zero price of any product. For ease of presentation, we overload on notation, and use $\mathcal{C}^{\mathcal{S}_f}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}$ as the resulting dynamic program, evaluated only over profitable states. The reason for choosing the term "profitable" will be made clear in our subsequent analysis. Intuitively, the collective revenue contribution of all non-profitable states will be shown to be negligible, meaning that such states can be overlooked by our dynamic program.

5.3. Concluding our analysis

Running time. As mentioned in Section 5.1, the current product index i and the size N of the assortment picked thus far take $O(n)$ values each, while the average price estimate $\tilde{P}^{\text{avg}}$ takes $O(\frac{n^3}{\epsilon} \cdot \log(\frac{np_{\max}}{p_{\min}}))$ possible values. Therefore, to show that our approach indeed results in an FPTAS, the next claim argues that the sum-of-utility-differences estimate $\tilde{U}^\Sigma$ takes polynomially many values over all profitable states. The proof appears in Appendix B.7.

Lemma 3. *$k_\ell\delta - \Delta \leq \tilde{U}^\Sigma \leq k_\ell\delta + \Delta$, for any profitable state $(i, N, \tilde{U}^\Sigma, \tilde{P}^{\text{avg}})$.*

Hence, over all profitable states, the sum-of-utility-differences estimate $\tilde{U}^\Sigma$ takes values in $[k_\ell\delta - \Delta, k_\ell\delta + \Delta] \cap \mathcal{U}_\epsilon$. Moreover, since the set $\mathcal{U}_\epsilon$ consists of integer multiples of δ, the possible number of such values is $O(\frac{\Delta}{\delta}) = O(\frac{n}{\epsilon} \cdot \log(\frac{n}{\epsilon} \cdot \frac{p_{\max}}{p_{\min}}))$, as $\Delta = \ln(\frac{4n}{\epsilon} \cdot \frac{p_{\max}}{p_{\min}})$ and $\delta = \frac{\epsilon}{2n}$. All in all, by taking into account the possible values for the additional parameters i, N, and $\tilde{P}^{\text{avg}}$, it follows that the number of profitable states is $O(\frac{n^6}{\epsilon^2} \cdot \log^2(\frac{np_{\max}}{p_{\min}}))$. In addition, it is worth mentioning that, by factoring out the term $e^{-k_\ell\delta}$, which is never computed explicitly, each of the scaled $\tilde{G}$-coefficients is associated with an exponent which is a multiple of δ in $[-\Delta, \Delta]$ and can therefore be evaluated in polynomial time. Summarizing the preceding discussion, we have just obtained the following theorem.

Theorem 5. *The objective value $\mathcal{C}^{\mathcal{S}_f}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}(\mathcal{S}_f)$, along with an assortment that attains this value, can be computed in* $\text{poly}(n, \frac{1}{\epsilon}, \log(\frac{p_{\max}}{p_{\min}}))$ *time.*

Approximation guarantee. To conclude our analysis, it remains to show that an optimal assortment $\hat{x}$ for the dynamic program $\mathcal{C}^{\mathcal{S}_f}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}$ constitutes a near-optimal assortment with respect to the original expected revenue function $\mathcal{R}$. Specifically, letting x^* be an optimal assortment for the latter function, we argue that $\mathcal{R}(\hat{x}) \geq (1-\epsilon)^4 \cdot \mathcal{R}(x^*)$. To this end, by inspecting the proof of Theorem 4, one can easily verify that it suffices to show that $\mathcal{C}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}(\hat{x}) \geq (1-\epsilon) \cdot \mathcal{C}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}(\bar{x})$, where $\bar{x}$ is an optimal assortment for $\mathcal{C}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}$. This fact is established via the next claim, whose proof is given in Appendix B.8.

Lemma 4. *$\mathcal{C}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}(\hat{x}) \geq (1-\epsilon) \cdot \mathcal{C}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}(\bar{x})$.*

6. Predictive Performance on Real-World Data

In this section, we present a case study that evaluates the predictive abilities of the Exponomial choice model relative to the MNL model on real-world choice data. Since both models make distinct, and yet equally parsimonious probabilistic assumptions on customers' random utilities, their goodness-of-fit on real-world data depends on the extent to which these assumptions are realistic in practice. Admittedly, one cannot expect any choice model to outperform alternative approaches on all data sets. That said, complementing the findings of Alptekinoglu and Semple (2016) and Berbeglia et al. (2018), we provide additional empirical evidence for the practical relevancy of the Exponomial choice model for choice prediction.

6.1. General MLE framework

Suppose that our data set consists of τ choice events, indexed by the set $T = \{1, \dots, \tau\}$. Each choice event $t \in T$ corresponds to the arrival of a single customer (user) who was offered an assortment of items $S_t \subseteq N = \{1, \dots, n\}$ and purchased (selected) the item $z_t \in S_t$. Furthermore, for each of the offered items $j \in S_t$, we denote by $X_{jt} \in \mathbb{R}^m$ the m-dimensional feature vector associated with this item, with the convention that $X_t = \{X_{jt} \in \mathbb{R}^m : j \in S_t\}$. For both the Exponomial and MNL choice models, we assume that the deterministic component of the utility function is a linear combination of feature values, weighted by an unknown coefficient vector β. For example, the ideal utility u_{jt} that customer t associates with product $j \in S_t$ is assumed to be given by $u_{jt} = \beta' X_{jt}$. The available choice data (CD) can therefore be fully specified through the set $\mathrm{CD} = \{(S_t, X_t, z_t) : t \in T\}$.

With this notation in place, the log-likelihood function is given by

$$\mathcal{LL}(\beta|\mathrm{CD}) = \log\left(\prod_{t=1}^{\tau} \pi(z_t, S_t, X_t, \beta)\right) = \sum_{t=1}^{\tau} \log\left(\pi(z_t, S_t, X_t, \beta)\right), \tag{11}$$

where $\pi(z_t, S_t, X_t, \beta)$ gives the likelihood of observing data point (z_t, S_t, X_t) under the presumed choice model of interest. Under the Exponomial model, $\pi(z_t, S_t, X_t, \beta)$ is as given in Section 2.3, while under the MNL model, we have $\pi(z_t, S_t, X_t, \beta) = \frac{e^{u_{z_t t}}}{1+\sum_{i \in S_t} e^{u_{it}}} = \frac{e^{\beta' X_{z_t t}}}{1+\sum_{i \in S_t} e^{\beta' X_{it}}}$. The general MLE problem of interest can therefore be formulated as

$$\max_{\beta} \mathcal{LL}(\beta|\mathrm{CD}). \tag{12}$$

The estimator obtained by solving (12) is the vector of coefficients β that maximizes the log-likelihood function given the available choice data. This optimization problem is convex when the choice probabilities are governed by either the Exponomial or MNL choice model (McFadden 1974, Alptekinoglu and Semple 2016), hence an optimal solution can be obtained through standard convex optimization tools. Additional details about our estimation methods and their implementation are provided in Appendix C.1.

6.2. Hotel booking data

Data description. We consider a publicly available data set that describes search results and bookings for hotels rooms on the Expedia platform. This data was released by Expedia in 2013 on the platform Kaggle, as part of a data science competition to improve their ranking engine[2]. In this setting, each choice event corresponds to a unique hotel search. As described below, we have access to the search query parameters, the hotels that were displayed to the user, and the hotel that was ultimately booked, if any.

Model specification. In this context, for each search query $t \in T$, the set S_t consists of the hotel options listed in the search results, and z_t is the option selected by the user, which corresponds either to a booked hotel $j \in S_t$ or to the no-purchase option. Additionally, the data set provides a large collection of potential explanatory features of each displayed hotel, such as the booking price, the webpage rank, the review scores, etc. A complete description of all 23 features included in X_t is given in Table 1, noting that we use a one-hot-encoding of the feature Display $\text{Position}_{jt}(k)$ within X_t. For both the Exponomial and MNL models, the deterministic component of the utility generated by each choice alternative $j \in S_t$ is expressed as a linear combination of the features:

$$u_{jt} = \beta_0 + \beta_1 \cdot \text{Gross Booking Price}_{jt} + \beta_2 \cdot \text{Display Position}_{jt}(0) + \cdots + \beta_{23} \cdot \text{Occupancy}_t \times \text{Trip}_t \ .$$

It is important to note that the no-purchase option has a utility of 0 and the coefficients $\beta_0, \ldots, \beta_{23}$ are constant across all hotel options. In other words, the choice alternatives are fully described by their feature vector.

Gross Booking Price_{jt}:	Nightly hotel rate for option $j \in S_t$.
Display $\text{Position}_{jt}(k)$:	Indicates whether the rank of option j is within $[5k, 5(k+1)]$, for each $k \in \{0, 1, \ldots, 7\}$.
Star Ratings_{jt} :	Star rating of the hotel from 1 to 5 (0 indicates no information) .
Avg Review Ratings_{jt} :	Average rating for the hotel in client reviews.
Location Score_{jt} :	Location score determined by Expedia.
Avg Hotel Price_j :	Average historical price of the hotel on a period preceding the experiment.
Is Promotion_{jt} :	Indicator of whether the hotel listing is under promotion.
Is Brand_{jt} :	Indicator of whether the hotel is part of a major hotel chain.
Trip_t:	Number of days before the beginning of the trip.
Stay_t:	Trip length (i.e., number of hotel nights).
# Rooms_t :	Number of rooms specified in the search query.
# Adults_t :	Number of adults specified in the search query.
# Children_t :	Number of children specified in the search query.
Occupancy_t :	Ratio between the number of travelers and the number of rooms.
Is Weekend_t :	Indicates whether the trip includes a weekend.
$\text{Price}_{jt} \times \text{Trip}_t$:	Interaction term between booking price and number of days before the trip.

Table 1 Search and hotel option features.

[2] See url: https://www.kaggle.com/c/expedia-hotel-recommendations.

Assumptions and data preprocessing. This data set allows for an almost ideal experimental setting, since the set of alternatives S_t displayed to each customer is precisely known for every search query. Furthermore, for each search query, the data set indicates whether the ranking of the displayed hotels was randomized or determined by the current Expedia ranking algorithm. We restrict our attention to searches of the former type in order to avoid endogeneity issues due to Expedia's algorithm. The empirical analysis of Ursu (2018) shows that price endogeneity is also not a concern in this setting, since the control variables explain nearly 80% of price variability. As such, we perform minimal preprocessing of the data, only dropping observations where at least one of the features of interest is missing. In addition, we discovered that for a small number of observations, the booking prices seem abnormally high and could correspond to corrupted entries. Consequently, we filter out any observation where the nightly room price is greater than $4000. Finally, we partition the Expedia searches by country websites (e.g. Expedia-UK), and restrict our attention to the 5 most popular ones by number of searches. These preprocessing steps yield 5 data sets of varying sizes.

6.3. Results

The summary results are described in Table 2, where we report three metrics that measure the out-of-sample predictive ability of the two fitted choice models. The data sets are chronologically split due to the time series nature of certain explanatory features: the training set is formed by the first 75% historical observations, while the test set is formed by the last 25% historical observations. We report the classification accuracy, the rank accuracy, and the normalized negative log-likelihood for each choice model. Classification accuracy is a standard metric corresponding to the fraction of observations where the actual choice decision is the alternative with the highest predicted purchase probability according to the choice model. The rank accuracy metric is defined by first computing the choice probability of each option in the choice set, sorting the alternatives in descending order of purchase probability, and finding the rank of the option that was selected. We use the convention that the option with the highest predicted purchase probability is assigned rank 1, the second highest is assigned rank 2, and so on. The value we report is the average rank of the selected option over all observations in the testing data.

Predictive performance. The Exponomial choice model outperforms the MNL model on our hotel booking data. We observe the most significant gaps in performance between the two choice models on the hotel booking data sets. More specifically, the negative log-likelihood is improved by 1-2%, while the predicted rank of the choice decisions is improved by 2-4%. It is worth pointing out that these differences in log-likelihood are significant in view of the typical magnitude of

Data set	Max. $\lvert S_t \rvert$	T	Classification MNL	Classification Exp	Rank MNL	Rank Exp	Neg. Log-Likelihood MNL	Neg. Log-Likelihood Exp
Expedia Site 1	38	87818	0.85	0.85	2.06	**2.01**	0.808	**0.795**
Expedia Site 2	36	14811	0.87	0.87	1.96	**1.89**	0.734	**0.720**
Expedia Site 3	36	9751	0.86	0.86	1.92	**1.86**	0.733	**0.722**
Expedia Site 4	32	7303	0.88	0.88	1.89	**1.83**	0.694	**0.682**
Expedia Site 5	36	5956	0.90	0.90	1.74	**1.68**	0.570	**0.558**

Table 2 Predictive power of the Exponomial and MNL models on real choice data.

variations observed in other empirical studies.[3] As one might expect, the two fitted models show little difference in terms of their respective classification accuracies over all five data sets. Indeed, since this metric is only concerned with predicting the highest utility item of each arriving customer or user, it can be less informative in settings where one alternative emerges by far as the most frequent (e.g., no-purchase in hotel booking). As a result, this metric is best suited for binary classification as opposed to choice modelling.

Analyzing the performance gap. In what follows, we wish to further understand in which choice environments the Exponomial model outperforms the MNL model and vice versa. In parallel to our work, Berbeglia et al. (2018) have conducted an empirical study that compares the predictive abilities of widely used choice models, including MNL and Exponomial, on a combination of synthetic and real-world data sets. The authors identify an important criterion for model selection: the degree of *consistency* in choice preferences. Customers' preferences are said to be consistent when the ground-truth model generating the choice data is well-approximated by a sparse rank-based model, i.e., a distribution over a small number of distinct rankings. Berbeglia et al. (2018) find that, in comparison to more complex choice models (e.g., Markov chain and mixtures of MNL), the predictive performance of the Exponomial and MNL models improves under lower levels of consistency in customers' preferences. Specifically, on synthetic training sets generated by distributions over 100 ranked lists, the Exponomial model emerges as the most predictive modeling approach out of those tested, whereas the MNL model outperforms a variant of the Markov chain model.

Hence, both choice models seemingly benefit from lower levels of consistency in preferences. However, the synthetic setting analyzed by Berbeglia et al. (2018) leaves us with two questions of practical interest: How can the consistency criterion be evaluated on real-world data sets? What does consistency imply for model selection amongst MNL and Exponomial? The notion of consistent preferences was defined by these authors in a synthetic experiment with respect to ground truth models taking the form of distributions over rankings. It is unclear how to extend this notion to real-world data sets, let alone measure it. Indeed, the class of rank-based models is difficult to

[3] For example, Şimşek and Topaloglu (2018) report 0-2% log-likelihood improvements in favor of the Markov Chain choice model over the MNL model on a hotel booking data set.

estimate from choice data; moreover, distinct distributions over rankings, with different levels of sparsity, could explain the given choice data equally well. Although conceptually appealing, the notion of consistent preferences cannot be converted into an easy-to-compute criterion.

To address the above shortcomings, we propose two metrics, *entropy* and *magnitude*, that are closely related to the notion of consistency, while being easy to compute on real-world data. We begin by defining these metrics assuming that the ground-truth choice probability $\hat{\pi}(i,t)$ is known at each choice event $t \in [\tau]$ for every item $i \in S_t$. In practice, while the ground truth choice probabilities are unknown, we can employ the predicted probabilities $\pi_{\mathrm{MNL}}(i,t)$ according to the fitted MNL model as a proxy. Consequently, the *entropy* associated with each choice event $t \in [\tau]$ is defined as $-\sum_{i\in S_t} \hat{\pi}(i,t) \cdot \log(\hat{\pi}(i,t))$. Similarly, we define the *magnitude* of each choice event $t \in [\tau]$ as $\sum_{i\in S_t} (\hat{\pi}(i,t))^2$. The entropy is an information-theoretic notion that quantifies the amount of information necessary to encode a given distribution. There is an intuitive connection between quantity of information and sparsity of distributions, and one would expect that rank-based models whose distribution is concentrated over fewer rankings (corresponding to highly consistent choice preferences) result in choice probability distributions of lower entropy and higher magnitude. This connection is theoretically established in Appendix C.2, where we show that entropy and magnitude are monotone with respect to the sparsity of the distribution over rankings for a broad class of generative settings.

Consequently, we formulate and empirically validate the following hypotheses:

H1 (entropy of preferences): The Exponomial model tends to outperform MNL when the entropy of the choice event is relatively small.

H2 (magnitude of choice probability): The Exponomial model tends to outperform MNL when the magnitude of the choice event is relatively large.

To test these hypotheses, we define the *performance gap* for a choice event $t \in [\tau]$ as the quantity $\log(\pi_{\mathrm{EXP}}(z_t,t)) - \log(\pi_{\mathrm{MNL}}(z_t,t)))$, where $\pi_{\mathrm{EXP}}(z_t,t)$ is the choice probability predicted by the fitted Exponomial model for the alternative picked at the choice event $t \in [\tau]$, and $\pi_{\mathrm{MNL}}(z_t,t)$ is the analogous quantity for the MNL model. Next, in Figure 1, we plot the performance gaps as a function of the entropy and magnitude on the largest Expedia data set (site 1). We restrict attention to choice events $t \in [\tau]$ for which $z_t \neq 0$, noting that our data set is imbalanced due to a majority of no-purchase events, whereby the performance gaps are very close to zero. Figure 1 indicates a negative relationship between our notion of entropy and the performance gaps between the Exponomial and MNL models, thereby corroborating hypothesis H1. Consistently with hypothesis H2, the magnitudes are positively correlated with the performance gap. We obtain nearly identical results on all other data sets and evaluate the statistical significance of these relationships; the reader is referred to Appendix C.3, where these additional results are presented. Based on the preceding discussion, we infer that the entropy and magnitude are correlated with the performance gap between the Exponomial and MNL choice models.

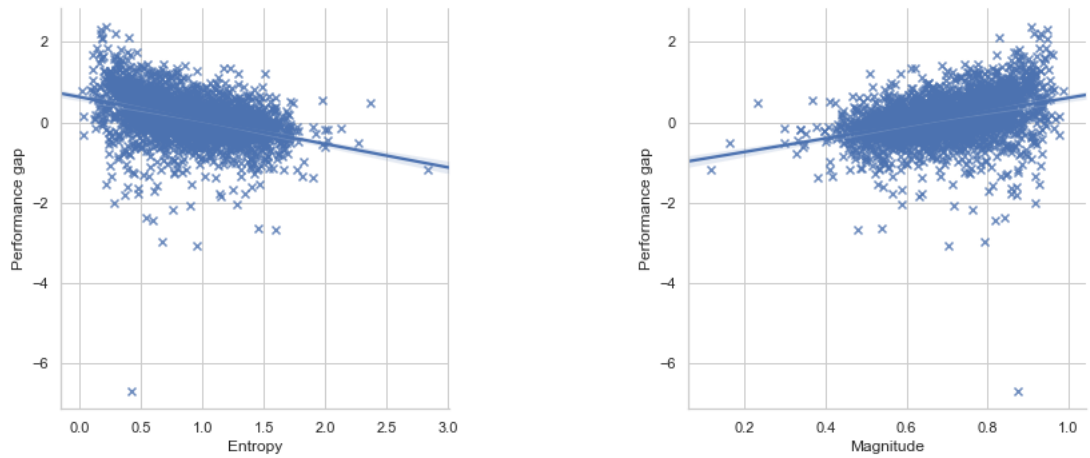
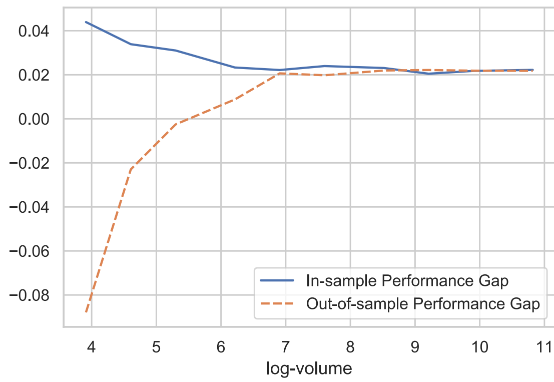


Figure 1 Performance gaps between the Exponomial and MNL choice models for the choice events $t \in [\tau]$, $z_t \neq 0$ on the Expedia site 1.

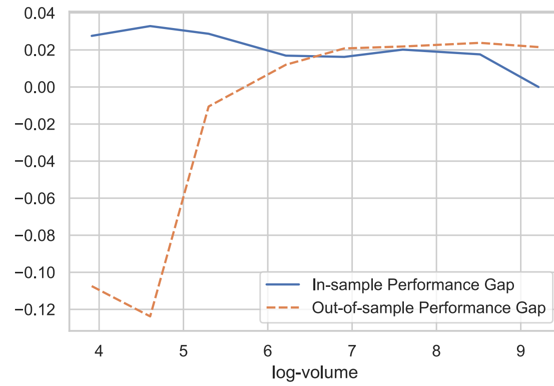
Effects of sample size. Another hypothesis substantiated by the work of Berbeglia et al. (2018) is that the relative predictive performance of the Exponomial choice model varies as a function of the amount of training data available:

H3 (effects of sample size): The Exponomial model tends to outperform other choice models in settings where the number of training observations is relatively small.

This hypothesis can be tested by varying the sample size of the training set utilized to fit the Exponomial and MNL models. Specifically, we measure the performance of the fitted models on randomly picked sub-samples of our original data sets. In Figure 2, we plot the average in-sample and out-of-sample performance gaps as a function of the volume of training data. This plot gives a more nuanced read on hypothesis H3. In terms of in-sample performance, the gap increases in favor of the Exponomial model as we decrease the sample size, meaning that this trend is in line with hypothesis H3. However, the out-of-sample performance gap evolves in the opposite direction; MNL achieves higher levels of out-of-sample accuracy than Exponomial when trained on smaller samples of the original training set. These observations suggest that the Exponomial model might present a higher risk of overfitting on small data sets. One potential explanation is that the experimental study of Berbeglia et al. (2018) focuses on synthetic data, generated from a rank-based model.



(a) Expedia site 1



(b) Expedia site 2

Figure 2 Normalized performance gaps as a function of the sample size τ.

7. Revenue Performance on Semi-Synthetic Data

In this section, we take our analysis one step further, by providing a series of experiments that investigate whether the recommended assortments derived from fitted Exponomial choice models are more profitable than those derived from fitted MNL and Markov Chain (MC) models. Using ordinal preference data, we evaluate the revenue generated by different assortment policies. Specifically, we employ a real-world data set comprising the preference lists of thousands of individuals over sushi items (Kamishima 2003). This setting comes in handy for counterfactual analysis, since we can precisely determine which item each customer would pick within any given assortment, assuming that their preference list is truthfully reported.

7.1. Experimental set-up

The sushi data set. We first construct a realistic ranking-based choice model, which is referred to in the remainder of this section as the *ground truth*. For this purpose, we employ real-world ordinal preference data over 10 sushi items, which were originally collected by Kamishima (2003) for the purpose of testing various collaborative-filtering-based recommendation algorithms. Specifically, the author surveyed 5,000 people, who were asked to rank 10 popular sushi items from most to least preferred. The end result of this survey is a publicly available[4] collection of 5,000 distinct full rankings of the 10 sushi items.

In what follows, let $N = \{1, \ldots, 10\}$ be the underlying set of 10 sushi items. Our ground truth model is constructed by sampling K rankings uniformly at random from the collection of 5,000 preference lists provided by Kamishima (2003). We insert the no-purchase option 0 in each preference list, while choosing its rank uniformly at random. Hence, for each $k \in [K]$, we view the resulting permutation σ_k over $N \cup \{0\}$ as a class of customers who purchase the highest ranked

[4] https://www.kamishima.net/sushi/

offered product in σ_k, including the no-purchase option. To generate the arrival probabilities of such customers, we independently sample K values $\beta_1, \ldots, \beta_K$ from an exponential distribution with mean 1, and then set $\lambda_k = \beta_k / \sum_{\kappa=1}^{K} \beta_\kappa$. This way, the arrival probabilities are uniformly sampled from the unit simplex. We vary the number of classes $K \in \{100, 250, 500\}$, and for each unique value of K, we generate 10 different ground choice models.

Generating sales data. Once a ground truth choice model has been determined, we then generate a stream of sales data, assuming that customers make purchasing decisions according to this model. This sales data set consists of $\tau = 2000$ customer arrivals, where the assortment S_t of products offered to customer t is determined by independently offering each product with probability 0.5. The product purchased by customer t is denoted as z_t. Hence, a single stream of sales data can be captured by the set $\mathrm{CD} = \{(S_t, z_t) : t = 1, \ldots, \tau\}$. For each stream, we generate 100 unique price vectors for the 10 sushi items. For each sushi item $i \in N$, we denote by p_i the so-called "normalized price" reported in the original data set of Kamishima (2003). Next, the revenue r_i of product i is picked uniformly at random from the interval $[0, p_i]$. For each stream of sales data, this approach yields a unique collection of 100 instances of the assortment optimization problem.

Fitting the models. Once we have generated the historical sales data, we fit each of the three models by directly maximizing their respective log-likelihood functions using MATLAB's constrained non-linear solver *fmincon*. Since we do not incorporate customer or product features, the Exponomial and MNL log-likelihoods are written solely as functions of the vector $U = (u_1, \ldots, u_n)$ of deterministic utilities. In other words, the MLE problem of interest can be formulated as

$$\max_U \sum_{t=1}^{\tau} \log\left(\pi(z_t, S_t, U)\right) \ ,$$

where $\pi(z_t, S_t, U)$ is the likelihood of observing data point (S_t, z_t) under the Exponomial or MNL model.

Under the MC choice model, substitution behavior is captured by transitions in a Markov chain, where each state corresponds to a different product. More specifically, each arriving customer begins in the state corresponding to his or her most preferred product. There is a specified distribution $\lambda \in [0,1]^n$ over the favorite products of the customer population. When this preferred product is unavailable, the customer transitions to another product according to the transition probabilities of the underlying Markov chain, which are specified by a transition matrix $P \in [0,1]^{n \times n}$. This process is repeated until he or she either reaches a state corresponding to an offered product, at which point the customer purchases this product, or reaches the state corresponding to the no-purchase option, at which point the customer leaves the store without making a purchase. Hence, the MLE problem under the MC model can be written as

$$\max_{\lambda, P} \sum_{t=1}^{\tau} \log\left(\pi(z_t, S_t, \lambda, P)\right), \tag{13}$$

where $\pi(z_t, S_t, \lambda, P)$ is the likelihood of observing data point (S_t, z_t) under the MC model. Unfortunately, the MC log-likelihood function is non-concave in its model parameters, however, several papers have shown that directly maximizing the log-likelihood function is nonetheless an efficacious approach (Şimşek and Topaloglu 2018). Along these lines, in an effort to avoid undesirable local optima, we solve the program (13) using MATLAB's *fmincon* solver. Since the MC model is a generalization of the MNL model (Blanchet et al. 2016), we plug in our estimate for the MNL parameters as the initial parameter vector. As a result, we ensure that the training log-likelihood of the fitted MC model is at least as large as that of the fitted MNL model.

Assessing the fitted models. Next, we test whether the fitted models can recommend profitable assortments. To this end, suppose that for a sales data set CD generated from the ground truth choice model GT, we have fitted Exponomial, MNL, and MC choice models, denoted as EP, MNL, and MC, respectively. For each choice model $\mathrm{CM} \in \{\mathrm{GT}, \mathrm{EP}, \mathrm{MNL}, \mathrm{MC}\}$, let $\pi_i^{\mathrm{CM}}(S)$ be the probability that product i is purchased under this model when the assortment S is offered. We measure the profitability of each choice model by computing the expected revenue of an optimal assortment it prescribes. To this end, for a fitted model CM, we compute the optimal recommended assortment $S^*(\mathrm{CM}) = \arg\max_{S \subseteq N} \sum_{i \in N} r_i \pi_i^{\mathrm{CM}}(S)$ by means of exhaustive enumeration, which is possible in our setting due to having only $n = 10$ products. We then evaluate the performance of these assortments by computing their expected revenue as per the ground truth choice model GT. In particular, for each fitted model $\mathrm{CM} \in \{\mathrm{EP}, \mathrm{MNL}, \mathrm{NL}\}$, we compute $R(\mathrm{CM}) = \sum_{i \in N} r_i \pi_i^{\mathrm{GT}}(S^*(\mathrm{CM}))$ in addition to the optimal expected revenue $R^* = \max_{S \subseteq N} \sum_{i \in N} r_i \pi_i^{\mathrm{GT}}(S)$. Consequently, for each instance of the assortment optimization problem, we store the percent optimality gap of the recommended assortment, given by $100 \cdot \frac{R^* - R(\mathrm{CM})}{R^*}$.

7.2. Results

Table 3 provides various statistics regarding the profitability of the recommended assortments over the three different values of K considered, recalling that this parameter stands for the number of customer types. In each row of this table, we report these statistics over all test instances with that particular value of K. In computing these values, we only consider test cases where $S^*(\mathrm{EP}) \neq S^*(\mathrm{MNL})$ or $S^*(\mathrm{EP}) \neq S^*(\mathrm{MC})$, i.e., we ignore cases for which all three fits recommend the same assortment, since these cases do not aid in assessing the relative revenue performance of the three models under consideration.

For each value of $K \in \{100, 250, 500\}$, there are hundreds of test cases ($\approx 25\%$) that satisfy this criterion. Columns two, three, and four of Table 3 give the average percentage optimality gap of the assortments recommended by the fitted Exponomial, MC, and MNL models, respectively, while columns five, six, and seven give the worst case optimality gaps. The final three columns specify

	Avg. % Opt. Gap			Max. % Opt. Gap			% Cases Best Rev.		
K	EP	MC	MNL	EP	MC	MNL	EP	MC	MNL
100	**2.77**	4.07	6.31	28.28	**26.75**	38.98	32.49	**54.62**	12.89
250	**1.41**	2.22	4.56	**16.15**	20.65	26.12	35.69	**49.12**	15.19
500	**0.94**	2.70	4.23	**18.82**	23.53	26.39	**47.16**	39.36	13.48

Table 3 Profitability of the recommended assortments under the Exponomial, MNL, and MC choice models.

the percentage of test cases in which the recommended assortment under each of the fits is the most profitable.

The results presented in Table 3 show that the expected revenues of the assortments recommended by the Exponomial fits are, on average, 3-4% more profitable than those recommended by the MNL fits, and 1-2% more profitable than those recommended by the MC fits. That said, for $K \in \{100, 250\}$, we observe that the MC fits recommend the most profitable assortment more frequently than the other two fits, as the latter three columns of Table 3 reveal. Furthermore, the average optimality gap of the assortments recommended by the Exponomial fits never exceeds 3%, which illustrates the impressive performance of the Exponomial model in absolute terms. Interestingly, we observe that as K increases from 100 to 500, the average percent optimality gap of the Exponomial and MNL fits drops uniformly by roughly 2%, while for the MC fits, this metric sees an uptick of 0.5% as K is increased from 250 to 500. Hence, as the number of customer classes that make-up the underlying ground truth choice model increases, we observe that the performance of the Exponomial fits actually improves, while the MC fits slightly deteriorates. All together, these results indicate that using the Exponomial choice model instead of the MNL or MC models can lead to significantly more profitable assortment decisions in certain cases.

Interestingly, while the revenue improvement of the assortments recommended by the Exponomial fits is substantial (2-4%), we observed that these fits barely improve upon the fitting accuracy of the MNL model, and are bested by the MC model in terms of fitting accuracy, as Table 4 indicates. These results suggest that the superior revenue performance of the Exponomial fits cannot be solely explained by the overall accuracy of the fitted model. Instead, one should more closely inspect the mechanisms at-play that determine the structure of each fit's recommended assortments. By analyzing the biases of the predicted choice probabilities, we find that the fitted MNL models over-estimate market share, which causes the assortment prescribed by each fit to be overly-sparse relative to the true optimal assortment. On the other hand, the fitted MC models occasionally underestimate market share, which causes the assortment prescribed by each fit to be overly-packed relative to the true optimal assortment. These two trends are discussed in detail in Appendix C.4.

	Log-Likelihood		
K	EP	MC	MNL
100	-2966.49	-2895.24	-2971.74
250	-3004.42	-2952.89	-3011.57
500	-2999.96	-2950.20	-3002.91

Table 4 Average training log-likelihood of the fitted Exponomial, MNL, and MC choice models.

8. Concluding Remarks

Cardinality and capacity constraints. As mentioned in Section 1.1, our approximation scheme actually operates in the more general cardinality-constrained setting, in which at most C products may be offered. To verify this statement, recall that the N-parameter within each state $(i, N, \tilde{U}^{\Sigma}, \tilde{P}^{\mathrm{avg}})$ specifies the number of products picked thus far, out of $1, \ldots, i-1$, including the no-purchase option ν. As the latter product is always offered, our recursive equations merely have to take into account that the current product i may be picked only when some capacity is still remaining (i.e., $N \leq C-1$). Based on this observation, our performance and running time analysis applies in its current form to the cardinality-constrained setting as well. In Appendix F, we explain how our dynamic programming approach can be further adapted to handle a general capacity (knapsack) constraint, in which case an FPTAS can be attained as well.

Open problems. Our work leads to numerous interesting directions for future research, from both theoretical and practical perspectives. As we have shown in Section 2.5, assortment optimization under the Exponomial model is NP-hard subject to totally-unimodular constraints, separating this setting from the Multinomial Logit choice model. However, it is still unclear whether the unconstrained or the cardinality-constrained versions are intractable or not; these seem challenging open questions. On a different note, it is worth mentioning that assortment optimization under an arbitrary mixture of Exponomial models can easily be shown to be $O(n^{1-\epsilon})$-hard to approximate via a straightforward reduction from the independent set problem (Håstad 1996). Another future direction of interest is that of incorporating the Exponomial choice model into additional decision problems in the context of revenue management, such as joint assortment and inventory management (Mahajan and van Ryzin 2001), discrete pricing (Sumida et al. 2021), or social-welfare assortment optimization (Shi 2015).

Additionally, Alptekinoğlu and Semple (2021) have recently introduced the Heteroscedastic Exponomial choice (HEC) model, which generalizes the traditional Exponomial model by allowing the variance of the exponentially-distributed error terms in the utility specification to be product-specific. Quite interestingly, the HEC model admits closed form expressions for its choice probabilities, yet to the best of our knowledge, its corresponding assortment problem remains unstudied. As such, a natural direction for future work is to examine whether the technical ideas behind our current FPTAS can be exploited to develop an FPTAS for the assortment problem under the HEC model.

Acknowledgements. The research of Danny Segev on this project was supported by Israel Science Foundation grants 148/10 and 1407/20.

References

Alptekinoglu A, Semple JH (2016) The Exponomial choice model: A new alternative for assortment and price optimization. *Operations Research* 64(1):79–93.

Alptekinoğlu A, Semple JH (2021) Heteroscedastic exponomial choice. *Operations Research* 69(3):841–858.

Aouad A, Farias V, Levi R, Segev D (2018a) The approximability of assortment optimization under ranking preferences. *Operations Research* 66(6):1661–1669.

Aouad A, Levi R, Segev D (2018b) Greedy-like algorithms for dynamic assortment planning under Multinomial Logit preferences. *Operations Research* 66(5):1321–1345.

Berbeglia G, Garassino A, Vulcano G (2018) A comparative empirical study of discrete choice models in retail operations. Management Science (forthcoming). Available online as SSRN report #3136816.

Bertsimas D, Misic VV (2019) Exact first-choice product line optimization. *Operations Research* 67(3):651–670.

Blanchet J, Gallego G, Goyal V (2016) A Markov chain approximation to choice modeling. *Operations Research* 64(4):886–905.

Bront J, Diaz IM, Vulcano G (2009) A column generation algorithm for choice-based network revenue management. *Operations Research* 57(3):769–784.

Daganzo C (1979) *Multinomial Probit: The Theory and Its Application to Demand Forecasting* (New York: Academic Press).

Davis J, Gallego G, Topaloglu H (2014) Assortment optimization under variants of the Nested Logit model. *Operations Research* 62(2):250–273.

Désir A, Goyal V, Zhang J (2022) Technical note – Capacitated assortment optimization: Hardness and approximation. *Operations Research* 70(2):893–904.

Feldman J, Topaloglu H (2015a) Bounding optimal expected revenues for assortment optimization under mixtures of Multinomial Logits. *Production and Operations Management* 24(10):1598–1620.

Feldman J, Topaloglu H (2015b) Capacity constraints across nests in assortment optimization under the nested logit model. *Operations Research* 63(4):812–822.

Ford Jr LR (1957) Solution of a ranking problem from binary comparisons. *The American Mathematical Monthly* 64(8P2):28–33.

Gallego G, Iyengar G, Phillips R, Dubey A (2004) Managing flexible products on a network. Technical Report TR-2004-01, Computational Optimization Research Center, Columbia University. Available online at: `http://www.corc.ieor.columbia.edu/reports/techreports/tr-2004-01.pdf`.

Gallego G, Li A, Truong V, Wang X (2016) Online bipartite matching with customer choice. Working paper. Available online at: `http://www.columbia.edu/~vt2196/ChoiceBasedRMORSubmission.pdf`.

Garey MR, Johnson DS (2002) *Computers and Intractability: A Guide to the Theory of NP-Completeness* (W. H. Freeman New York).

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

Håstad J (1996) Clique is hard to approximate within $n^{1-\epsilon}$. *Proceedings of the 37th Annual Symposium on Foundations of Computer Science*, 627–636.

Jagabathula S (2014) Assortment optimization under general choice. Working paper. Available online as SSRN report #2512831.

Jagabathula S, Subramanian L, Venkataraman A (2020) A conditional gradient approach for nonparametric estimation of mixing distributions. *Management Science* 66(8):3635–3656.

Kallus N, Udell M (2020) Dynamic assortment personalization in high dimensions. *Operations Research* 68(4):1020–1037.

Kamishima T (2003) Collaborative filtering: Recommendation based on order responses. *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 583–588.

Karp RM (1972) Reducibility among combinatorial problems. *Complexity of Computer Computations*, 85–103 (Springer).

Kök AG, Fisher ML, Vaidyanathan R (2015) Assortment planning: Review of literature and industry practice. Agrawal N, Smith SA, eds., *Retail Supply Chain Management: Quantitative Models and Empirical Studies*, 175–236 (Boston, MA: Springer US).

Luce RD (1959) Individual choice behavior: A theoretical analysis. *Frontiers in Econometrics* 2:105–142.

Mahajan S, van Ryzin G (2001) Stocking retail assortment under dynamic consumer substitution. *Operations Research* 49(3):334–351.

McFadden D (1974) Conditional Logit analysis of qualitative choice behavior. *Frontiers in Econometrics* 2:105–142.

Rusmevichientong P, Shen ZJM, Shmoys DB (2010) Dynamic assortment optimization with a Multinomial Logit choice model and capacity constraint. *Operations Research* 58(6):1666–1680.

Rusmevichientong P, Shmoys D, Tong C, Topaloglu H (2014) Assortment optimization under the Multinomial Logit model with random choice parameters. *Production and Operations Management* 23(11):2023–2039.

Rusmevichientong P, Sumida M, Topaloglu H (2020) Dynamic assortment optimization for reusable products with random usage durations. *Management Science* 66(7):2820–2844.

Rusmevichietong P, Jagabathula S (2017) A nonparametric joint assortment and price choice model. *Management Science* 60(6):1532–1551.

Schrijver A (1998) *Theory of Linear and Integer Programming* (John Wiley & Sons).

Shaked M, Shanthikumar JG (2007) *Stochastic Orders* (Springer Science & Business Media).

Shi P (2015) Guiding school-choice reform through novel applications of operations research. *Interfaces* 45(2):117–132.

Şimşek AS, Topaloglu H (2018) An expectation-maximization algorithm to estimate the parameters of the Markov chain choice model. *Operations Research* 66(3):748–760.

Sumida M, Gallego G, Rusmevichientong P, Topaloglu H, Davis J (2021) Revenue-utility tradeoff in assortment optimization under the Multinomial Logit model with totally unimodular constraints. *Management Science* 67(5):2845–2869.

Talluri K, van Ryzin G (2004) Revenue management under a general discrete choice model of consumer behavior. *Management Science* 50(24):15–33.

Thurston L (1927) A law of comparative judgement. *Psychological Review* 34:273–286.

Train K (2009) *Discrete Choice Methods with Simulation* (Cambridge University Press).

Ursu RM (2018) The power of rankings: Quantifying the effect of rankings on online consumer search and purchase decisions. *Marketing Science* 37(4):530–552.

Vulcano G, van Ryzin G, Ratliff R (2012) Estimating primary demand for substitutable products from sales transaction data. *Operations Research* 60(2):313–334.

Zhang H, Rusmevichientong P, Topaloglu H (2020) Assortment optimization under the paired combinatorial Logit model. *Operations Research* 68(3):741–761.

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 Companion

Appendix A: Hardness Results

A.1. Proof of Theorem 1

Our proof is based on a reduction from the maximum knapsack problem with an exact cardinality constraint, to which we refer as EC-knapsack. An instance of this decision problem consists of a collection of n items, where each item i has a value of v_i and a weight w_i. In addition, we are given an integer parameter k, that specifies the desired cardinality of the chosen subset, as well as an upper bound B_w on the total weight picked and a lower bound B_v on the total value gained. The question is whether there exists a subset of items $I \subseteq [n]$ of cardinality exactly k such that $w(I) = \sum_{i \in I} w_i \leq B_w$ and $v(I) = \sum_{i \in I} v_i \geq B_v$? It is easy to verify that EC-knapsack is NP-complete via a straightforward reduction from the standard maximum knapsack problem (Garey and Johnson 2002, Problem MP9, pg. 247).

Given an EC-knapsack instance, we map it to a corresponding instance of the capacity-constrained assortment optimization problem as follows:

- For each item $i \in [n]$ there is a corresponding product, with price $p_i = 1 + \frac{v_i}{k(k+2)V}$, utility $u_i = 1$, and size $s_i = 1 + \frac{w_i}{(k+1)W}$. Here, $V = v([n])$ and $W = w([n])$.
- The no-purchase option is represented by product $n+1$, with price $p_{n+1} = 0$, utility $u_{n+1} = 1$, and size $s_{n+1} = 0$.
- Except for the no-purchase option, which is always picked, feasible assortments are those with a total size of at most $B = k + \frac{B_w}{(k+1)W}$.

Letting $S^* \subseteq [n+1]$ be an optimal set of products for the resulting capacity-constrained assortment optimization instance, we first observe that $|S^*| = k+1$, since:

- If $|S^*| \geq k+2$, its total size is

$$s(S^*) = \sum_{i \in S^* \cap [n]} \left(1 + \frac{w_i}{(k+1)W}\right) \geq |S^*| - 1 \geq k+1 > k + \frac{B_w}{(k+1)W} = B \ ,$$

meaning that this case is impossible.

- If $|S^*| \leq k$, its expected revenue can be upper-bounded as follows:

$$\mathcal{R}(S^*) = \frac{1}{|S^*|} \cdot \sum_{i \in S^*} p_i = \frac{1}{|S^*|} \cdot \left(|S^*| - 1 + \frac{\sum_{i \in S^* \cap [n]} v_i}{k(k+2)V}\right) \leq 1 - \frac{1}{k} + \frac{1}{k(k+2)} \ ,$$

where the first equality holds since all products have identical utilities, implying that the choice probabilities of products in S^* are uniform. However, any feasible assortment S with $|S| = k+1$ has an expected revenue of

$$\mathcal{R}(S) = \frac{1}{|S|} \cdot \sum_{i \in S} p_i \geq \frac{|S| - 1}{|S|} = 1 - \frac{1}{k+1} > \mathcal{R}(S^*) \ ,$$

so this case is impossible as well.

With this observation in place, we now argue that our EC-knapsack input is a YES instance if and only if $\mathcal{R}(S^*) \geq \frac{k}{k+1} + \frac{B_v}{k(k+1)(k+2)V}$:

- YES instance $\Rightarrow \mathcal{R}(S^*) \geq \frac{k}{k+1} + \frac{B_v}{k(k+1)(k+2)V}$. Suppose that the subset $I \subseteq [n]$ satisfies $|I| = k$, $w(I) \leq B_w$, and $v(I) \geq B_v$. Then, the assortment $S_I = I \cup \{n+1\}$ has a total size of $s(S_I) = k + \frac{w(I)}{(k+1)W} \leq k + \frac{B_w}{(k+1)W} = B$, meaning that it is feasible. In addition, the expected revenue of this assortment is

$$\mathcal{R}(S_I) = \frac{1}{|S_I|} \cdot \sum_{i \in S_I} p_i = \frac{k}{k+1} + \frac{v(I)}{k(k+1)(k+2)V} \geq \frac{k}{k+1} + \frac{B_v}{k(k+1)(k+2)V} \ .$$

Hence, due to the optimality of S^*, we have $\mathcal{R}(S^*) \geq \mathcal{R}(S_I) \geq \frac{k}{k+1} + \frac{B_v}{k(k+1)(k+2)V}$.

- $\mathcal{R}(S^*) \geq \frac{k}{k+1} + \frac{B_v}{k(k+1)(k+2)V} \Rightarrow$ YES instance. The arguments are nearly identical in the opposite direction, and are therefore omitted.

A.2. Proof of Theorem 2

Our proof is based on a reduction from the set partition problem (Karp 1972). Here, we are given n positive integers $a_1, \ldots, a_n$, and the question is whether these numbers can be partitioned into two equal-sum sets. In other words, letting $\mathcal{A} = a([n])$, we would like to decide whether there exists a subset $I \subseteq [n]$ such that $a(I) = \mathcal{A}/2$.

Step 1: Modified instance. It is easy to verify that the set partition problem remains NP-complete even with the additional constraint of dividing the given numbers into sets of equal cardinality. Now, for every $i \in [n]$, let $b_i = \mathcal{A} + a_i$ and $\mathcal{B} = b([n]) = (n+1)\mathcal{A}$. In Appendix A.3, we prove the next auxiliary claim.

Lemma EC.1. *For any subset $I \subseteq [n]$, we have $|I| = n/2$ and $a(I) = \mathcal{A}/2$ if and only if $b(I) = \mathcal{B}/2$.*

Step 2: The reduction. We construct an instance of the TU-constrained assortment optimization problem under the Exponomial model as follows:

- For every $i \in [n]$, there are two products, i^+ and i^-, with prices $p_{i^+} = b_i/\mathcal{B}$ and $p_{i^-} = 0$. The utilities of these products are $u_{i^+} = 0$ and $u_{i^-} = 2b_i/\mathcal{B}$.
- In addition, there are $N = 80n^3$ dummy products, with price 0 and utility 0. We denote the set of dummy products by $\mathcal{D}$.
- The no-purchase option ν has price 0 and utility $-\infty$.
- TU constraints: For every $i \in [n]$, if product i^+ is picked, we also have to pick product i^- as well as all N dummy products in $\mathcal{D}$. It is easy to verify that these constraints can be collectively expressed as $Ax^S \leq 0$, where the matrix A has a single appearance of $+1$ and -1 in each row, with all other entries being 0. Such matrices, which can be thought of as incidence matrices of directed graphs, are known to be Totally Unimodular (see, for example, Schrijver (1998)).

The expected revenue function. Despite the rather cumbersome expressions for the choice probabilities under the Exponomial model, the above construction enables us to obtain a succinct representation for the expected revenue of a given assortment, as stated in the next lemma.

Lemma EC.2. *Let x be a feasible assortment and let $I = \{i \in [n] : x_{i^+} = 1\}$. Then,*

$$\mathcal{R}(x) = \frac{1}{N + 2 \cdot |I|} \cdot \frac{b(I)}{\mathcal{B}} \cdot e^{-2b(I)/\mathcal{B}} \ .$$

Proof. Letting $K = |I|$, we denote $I = \{i_1, \dots, i_K\}$ and assume without loss of generality that $b_{i_1} \geq \cdots \geq b_{i_K}$. Due to the Totally Unimodular constraints, the collection of products picked, by order of non-increasing utilities, is given by $i_1^-, \dots, i_K^-, \leftarrow \mathcal{D} \rightarrow, i_1^+, \dots, i_K^+, \nu$. As a result, by the expected revenue representation (1), since $i_1^-, \dots, i_K^-$ and the dummy products all have zero prices,

$$\begin{aligned}\mathcal{R}(x) &= \sum_{k=1}^{K} G_{i_k^+}(x) \cdot \left(p_{i_k^+} - \frac{1}{X(i_k^+ - 1)} \cdot \sum_{j=1}^{i_k^+ - 1} p_j x_j \right) \\ &= \sum_{k=1}^{K} \frac{1}{K+N+k} \cdot e^{-2b(I)/\mathcal{B}} \cdot \left(\frac{b_{i_k}}{\mathcal{B}} - \frac{1}{K+N+k-1} \cdot \sum_{j=1}^{k-1} \frac{b_{i_j}}{\mathcal{B}} \right) \\ &= \frac{1}{N+2K} \cdot \frac{b(I)}{\mathcal{B}} \cdot e^{-2b(I)/\mathcal{B}} \ .\end{aligned}$$

□

Reduction guarantees. We are now ready to relate between original set partition instances and their mapped assortment optimization instances. Specifically, in Lemmas EC.3 and EC.4 below, we show that YES instances of set partition lead to an optimal expected revenue of at least $\frac{1}{2e \cdot (N+n)}$, whereas NO instances lead to an optimum of at most $\frac{1}{2e \cdot (N+n)} - \min\{\frac{1}{20N\mathcal{B}^2}, \frac{1}{70000Nn^4}\}$. It is imperative to note that the latter gap between YES and NO instances is indeed necessary, since in order to specify the revenue threshold that results from our reduction, the precise term $\frac{1}{2e \cdot (N+n)}$ cannot be used due to being an irrational number. For this purpose, a gap of $\frac{1}{\text{poly}(N,n,\mathcal{B})}$ shows that polynomially many leading digits of the threshold $\frac{1}{2e \cdot (N+n)}$ are sufficient.

Lemma EC.3 **(YES instances)**. *Suppose there exists a subset $I \subseteq [n]$ with $b(I) = \mathcal{B}/2$. Then, the expected revenue of the optimal assortment x^* satisfies $\mathcal{R}(x^*) \geq \frac{1}{2e \cdot (N+n)}$.*

Proof. Let x_I be the assortment consisting of the products $\{i^+\}_{i \in I} \cup \{i^-\}_{i \in I} \cup \mathcal{D}$, which is clearly a feasible one. By Lemma EC.2, the expected revenue of this assortment is

$$\mathcal{R}(x_I) = \frac{1}{N + 2 \cdot |I|} \cdot \frac{b(I)}{\mathcal{B}} \cdot e^{-2b(I)/\mathcal{B}} = \frac{1}{2e \cdot (N+n)} \ ,$$

where the last equality holds since $b(I) = \mathcal{B}/2$ and since $|I| = n/2$, by Lemma EC.1. □

Lemma EC.4 **(NO instances)**. *Suppose $b(I) \neq \mathcal{B}/2$ for every subset $I \subseteq [n]$. Then, the expected revenue of the optimal assortment x^* satisfies*

$$\mathcal{R}(x^*) \leq \frac{1}{2e \cdot (N+n)} - \min\left\{ \frac{1}{20N\mathcal{B}^2}, \frac{1}{70000Nn^4} \right\} \ .$$

Proof. Let $I^* = \{i \in [n] : x^*_{i^+} = 1\}$. To prove the claim, we consider two cases, depending on the cardinality of I^*.

Case 1: $|I^*| = n/2$. Here, by Lemma EC.2 we have $\mathcal{R}(x^*) = \frac{1}{N+n} \cdot \frac{b(I^*)}{\mathcal{B}} \cdot e^{-2b(I^*)/\mathcal{B}}$. Letting $\psi(z) : [0,1] \rightarrow \mathbb{R}$ be the function defined by $\psi(z) = ze^{-2z}$, it follows that $\mathcal{R}(x^*) = \frac{1}{N+n} \cdot \psi(\hat{z})$ for some $\hat{z} \in [0,1]$ satisfying $|\hat{z} - \frac{1}{2}| \geq \frac{1}{\mathcal{B}}$, since $b(I^*) \neq \mathcal{B}/2$ and since $b(I^*)$ is an integer. Consequently,

$$\mathcal{R}(x^*) = \frac{1}{N+n} \cdot \psi(\hat{z})$$

$$\leq \frac{1}{N+n} \cdot \left(\frac{1}{2e} - \frac{|\hat{z}-1/2|^2}{10} \right)$$
$$\leq \frac{1}{2e \cdot (N+n)} - \frac{1}{20N\mathcal{B}^2} \ ,$$

where the first inequality follows from Claim EC.1 below, whose proof is given in Appendix A.4, and the second inequality holds since $|\hat{z}-\frac{1}{2}| \geq \frac{1}{\mathcal{B}}$ and $N \geq n$.

Claim EC.1. *$\psi(\hat{z}) \leq \frac{1}{2e} - \frac{|\hat{z}-1/2|^2}{10}$, for every $\hat{z} \in [0,1]$.*

Case 2: $|I^*| \neq n/2$. In this case, by Lemma EC.2 we have $\mathcal{R}(x^*) = \frac{1}{N+2\cdot|I^*|} \cdot \frac{b(I^*)}{\mathcal{B}} \cdot e^{-2b(I^*)/\mathcal{B}}$. Similar to the previous case, $\mathcal{R}(x^*)$ and $\psi(\cdot)$ can be related as stated in the next claim, whose proof is given in Appendix A.5.

Claim EC.2. *$\mathcal{R}(x^*) = \frac{1}{N+2\cdot|I^*|} \cdot \psi(\hat{z})$ for some $\hat{z} \in [0,1]$ satisfying $|\hat{z}-\frac{1}{2}| \geq \frac{1}{4n}$.*

Consequently,

$$\begin{aligned}
\mathcal{R}(x^*) &= \frac{1}{N+2\cdot|I^*|} \cdot \psi(\hat{z}) \\
&\leq \frac{N+n}{N+2\cdot|I^*|} \cdot \frac{1}{N+n} \cdot \left(\frac{1}{2e} - \frac{|\hat{z}-1/2|^2}{10} \right) \\
&\leq \frac{N+n}{N+2\cdot|I^*|} \cdot \frac{1}{N+n} \cdot \left(\frac{1}{2e} - \frac{1}{160n^2} \right) \\
&\leq \left(1+\frac{n}{N}\right) \cdot \frac{1}{N+n} \cdot \left(1 - \frac{1}{80n^2}\right) \cdot \frac{1}{2e} \\
&= \left(1+\frac{1}{80n^2}\right) \cdot \frac{1}{2e\cdot(N+n)} \cdot \left(1-\frac{1}{80n^2}\right) \\
&= \left(1-\frac{1}{6400n^4}\right) \cdot \frac{1}{2e\cdot(N+n)} \\
&\leq \frac{1}{2e\cdot(N+n)} - \frac{1}{70000Nn^4} \ ,
\end{aligned}$$

where the first inequality follows from Claim EC.1, the second inequality holds since $|\hat{z}-\frac{1}{2}| \geq \frac{1}{4n}$ by Claim EC.2, and the second equality holds since $N = 80n^3$. $\square$

A.3. Proof of Lemma EC.1

First, if $|I| = n/2$ and $a(I) = \mathcal{A}/2$ then

$$b(I) = |I| \cdot \mathcal{A} + a(I) = \frac{(n+1)\mathcal{A}}{2} = \frac{\mathcal{B}}{2} \ .$$

Conversely, when $b(I) = \mathcal{B}/2$ we must have $|I| = n/2$. Otherwise, there are two cases:

- $|I| \leq \frac{n}{2} - 1$, in which case $b(I) \leq (\frac{n}{2}-1) \cdot \mathcal{A} + \mathcal{A} = \frac{n\mathcal{A}}{2} < \frac{\mathcal{B}}{2}$.
- $|I| \geq \frac{n}{2} + 1$, in which case $b(I) \geq (\frac{n}{2}+1) \cdot \mathcal{A} > \frac{\mathcal{B}}{2}$.

In addition, since we have just shown that $|I| = n/2$,

$$a(I) = b(I) - |I| \cdot \mathcal{A} = \frac{\mathcal{B} - n\mathcal{A}}{2} = \frac{\mathcal{A}}{2} \ .$$

A.4. Proof of Claim EC.1

When $\hat{z} = \frac{1}{2}$, the desired inequality is actually tight, and we therefore consider two different settings, with $\hat{z} < \frac{1}{2}$ and $\hat{z} > \frac{1}{2}$. Elementary calculus arguments show that the function $\psi(z)$ is concave over the interval $[0,1]$, with a unique maximizer at $z = 1/2$. First, in order to deal with the case $\hat{z} < \frac{1}{2}$, note that the (differentiable) function ψ lies below all of its tangents, meaning in particular that for the tangent at $\frac{\hat{z}+1/2}{2}$ we have $\psi(z) \leq \psi(\frac{\hat{z}+1/2}{2}) + \psi'(\frac{\hat{z}+1/2}{2}) \cdot (z - \frac{\hat{z}+1/2}{2})$. By substituting $z = \hat{z}$, it follows that

$$\begin{aligned}
\psi(\hat{z}) &\leq \underbrace{\psi\left(\frac{\hat{z}+1/2}{2}\right)}_{\leq \psi(1/2)=1/(2e)} + \psi'\left(\frac{\hat{z}+1/2}{2}\right) \cdot \left(\hat{z} - \frac{\hat{z}+1/2}{2}\right) \\
&\leq \frac{1}{2e} - \frac{|\hat{z}-1/2|}{2} \cdot \left(e^{-2z} \cdot (1-2z)\right)\Big|_{z=\frac{\hat{z}+1/2}{2}} \\
&\leq \frac{1}{2e} - \frac{|\hat{z}-1/2|}{2} \cdot \left(\underbrace{e^{-(\hat{z}+1/2)}}_{\geq e^{-3/2} \geq 1/5} \cdot \left(\frac{1}{2} - \hat{z}\right)\right) \\
&\leq \frac{1}{2e} - \frac{|\hat{z}-1/2|^2}{10} \ .
\end{aligned}$$

Similarly, for the case $\hat{z} > \frac{1}{2}$, we have

$$\begin{aligned}
\psi(\hat{z}) &\leq \underbrace{\psi\left(\frac{\hat{z}+1/2}{2}\right)}_{\leq \psi(1/2)=1/(2e)} + \psi'\left(\frac{\hat{z}+1/2}{2}\right) \cdot \left(\hat{z} - \frac{\hat{z}+1/2}{2}\right) \\
&\leq \frac{1}{2e} + \frac{|\hat{z}-1/2|}{2} \cdot \left(e^{-2z} \cdot (1-2z)\right)\Big|_{z=\frac{\hat{z}+1/2}{2}} \\
&= \frac{1}{2e} + \frac{|\hat{z}-1/2|}{2} \cdot \left(e^{-(\hat{z}+1/2)} \cdot \underbrace{\left(\frac{1}{2} - \hat{z}\right)}_{=-|\hat{z}-1/2|}\right) \\
&= \frac{1}{2e} - \frac{|\hat{z}-1/2|^2}{2} \cdot \underbrace{e^{-(\hat{z}+1/2)}}_{\geq e^{-3/2} \geq 1/5} \\
&\leq \frac{1}{2e} - \frac{|\hat{z}-1/2|^2}{10} \ .
\end{aligned}$$

A.5. Proof of Claim EC.2

To establish the desired claim, we distinguish between two cases, depending on the cardinality of I^*:

- When $|I^*| < n/2$, we have

$$\begin{aligned}
b(I^*) &= |I^*| \cdot \mathcal{A} + a(I^*) \\
&\leq \left(\frac{n}{2} - 1\right) \cdot \mathcal{A} + \mathcal{A} \\
&= \frac{(n+1)\mathcal{A}}{2} - \frac{\mathcal{A}}{2} \\
&= \frac{\mathcal{B}}{2} - \frac{\mathcal{A}}{2} \\
&\leq \left(\frac{1}{2} - \frac{1}{4n}\right) \mathcal{B} \ .
\end{aligned}$$

- When $|I^*| > n/2$,

$$\begin{aligned} b(I^*) &= |I^*| \cdot \mathcal{A} + a(I^*) \\ &\geq \left(\frac{n}{2}+1\right)\mathcal{A} \\ &= \frac{(n+1)\mathcal{A}}{2} + \frac{\mathcal{A}}{2} \\ &\geq \left(\frac{1}{2}+\frac{1}{4n}\right)\mathcal{B} \ . \end{aligned}$$

Appendix B: Additional Proofs

B.1. Proof of Lemma 1

To prove the first part of this claim, $\text{avg}_{i-1}(x) \geq \frac{1}{X(i-1)} \cdot \sum_{j=1}^{i-1} p_j x_j$, we proceed by induction on i. The base case $i=1$ is obvious, since $\text{avg}_0(x) = 0$ by definition. For the general case, where $i \geq 2$, we have

$$\begin{aligned} \text{avg}_{i-1}(x) &= \mathcal{P}_\epsilon\left(\frac{X(i-2)\cdot \text{avg}_{i-2}(x) + p_{i-1}x_{i-1}}{X(i-1)}\right) \\ &\geq \frac{X(i-2)\cdot \text{avg}_{i-2}(x) + p_{i-1}x_{i-1}}{X(i-1)} \\ &\geq \frac{1}{X(i-1)} \cdot \left(\sum_{j=1}^{i-2} p_j x_j + p_{i-1}x_{i-1}\right) \\ &= \frac{1}{X(i-1)} \cdot \sum_{j=1}^{i-1} p_j x_j \ , \end{aligned}$$

where the second inequality follows from the induction hypothesis.

The second part, $\text{avg}_{i-1}(x) \leq \frac{(1+\theta)^{i-1}}{X(i-1)} \cdot \sum_{j=1}^{i-1} p_j x_j$, can be derived by induction in a similar way. Again, the base case $i=1$ is obvious, since $\text{avg}_0(x) = 0$. For the general case, where $i \geq 2$,

$$\begin{aligned} \text{avg}_{i-1}(x) &= \mathcal{P}_\epsilon\left(\frac{X(i-2)\cdot \text{avg}_{i-2}(x) + p_{i-1}x_{i-1}}{X(i-1)}\right) \\ &\leq (1+\theta)\cdot\left(\frac{X(i-2)\cdot \text{avg}_{i-2}(x) + p_{i-1}x_{i-1}}{X(i-1)}\right) \\ &\leq \frac{1+\theta}{X(i-1)} \cdot \left((1+\theta)^{i-2} \cdot \sum_{j=1}^{i-2} p_j x_j + p_{i-1}x_{i-1}\right) \\ &\leq \frac{(1+\theta)^{i-1}}{X(i-1)} \cdot \sum_{j=1}^{i-1} p_j x_j \ , \end{aligned}$$

where the second inequality follows from the induction hypothesis.

B.2. Proof of Theorem 3

To establish the desired result, we argue that

$$\mathcal{R}(\tilde{x}) \geq \mathcal{C}_{\mathcal{P}_\epsilon}(\tilde{x}) \geq \mathcal{C}_{\mathcal{P}_\epsilon}(x^*) \geq (1-\epsilon)\cdot \mathcal{R}(x^*) \ .$$

Here, the first inequality is implied by Claim EC.3, the middle inequality follows from the optimality of $\tilde{x}$ for $\mathcal{C}_{\mathcal{P}_\epsilon}$, and the third inequality is proven in Claim EC.4.

Claim EC.3. *$\mathcal{R}(x) \geq \mathcal{C}_{\mathcal{P}_\epsilon}(x)$, for any assortment x.*

Proof. To this end, note that

$$\begin{aligned}
\mathcal{C}_{\mathcal{P}_\epsilon}(x) &= \sum_{i=1}^{n} G_i(x) \cdot \left(p_i - \mathrm{avg}_{i-1}(x)\right) \cdot x_i \\
&\leq \sum_{i=1}^{n} G_i(x) \cdot \left(p_i - \frac{1}{X(i-1)} \cdot \sum_{j=1}^{i-1} p_j x_j\right) \cdot x_i \\
&= \mathcal{R}(x) \ ,
\end{aligned}$$

where the inequality above follows from Lemma 1. □

Claim EC.4. $\mathcal{C}_{\mathcal{P}_\epsilon}(x^*) \geq (1-\epsilon) \cdot \mathcal{R}(x^*)$.

Proof. In order to obtain a lower bound on $\mathcal{C}_{\mathcal{P}_\epsilon}(x^*)$, note that

$$\begin{aligned}
\mathcal{C}_{\mathcal{P}_\epsilon}(x^*) &= \sum_{i=1}^{n} G_i(x^*) \cdot \left(p_i - \mathrm{avg}_{i-1}(x^*)\right) \cdot x_i^* \\
&\geq \sum_{i=1}^{n} G_i(x^*) \cdot \left(p_i - \frac{(1+\theta)^{i-1}}{X^*(i-1)} \cdot \sum_{j=1}^{i-1} p_j x_j^*\right) \cdot x_i^* \\
&\geq \sum_{i=1}^{n} G_i(x^*) \cdot \left(p_i - \frac{1+\epsilon/n^2}{X^*(i-1)} \cdot \sum_{j=1}^{i-1} p_j x_j^*\right) \cdot x_i^* \\
&\geq \mathcal{R}(x^*) - \frac{\epsilon}{n^2} \cdot \sum_{i=1}^{n} \sum_{j=1}^{i-1} G_i(x^*) \cdot p_j x_j^* \\
&\geq \mathcal{R}(x^*) - \frac{\epsilon}{n^2} \cdot \sum_{i=1}^{n} \sum_{j=1}^{i-1} G_j(x^*) \cdot p_j x_j^* \\
&\geq (1-\epsilon) \cdot \mathcal{R}(x^*) \ .
\end{aligned}$$

Here, the first equality and the subsequent inequality follow from (7) and Lemma 1, respectively. The second inequality holds since $(1+\theta)^{i-1} \leq e^{n\theta} = e^{\epsilon/(2n^2)} \leq 1 + \frac{\epsilon}{n^2}$, recalling that $\theta = \frac{\epsilon}{2n^3}$. The third inequality follows from our alternative representation of the expected revenue in (1). The fourth inequality is obtained by observing that $G_i(x) \leq G_j(x)$ when $i > j$, for any assortment x. The last inequality follows from the next technical claim, whose proof appears in Appendix B.3.

Claim EC.5. *$G_i(x) \cdot p_i x_i \leq 2\mathcal{R}(x^*)$, for any assortment x and for every product $i \in [n]$.*

□

B.3. Proof of Claim EC.5

When product i is the no-purchase option ν, or is simply not stocked by the assortment x, then $G_i(x) \cdot p_i x_i = 0$ and the claim is trivially satisfied. We therefore consider the interesting case, where $i \neq \nu$ and $x_i = 1$. In this setting, let $x_{[i]}$ be the assortment that stocks only product i and the no-purchase option ν. Since $x_{[i]} \leq x$, it is easy to verify that $G_i(x_{[i]}) \geq G_i(x)$, implying that

$$G_i(x) \cdot p_i x_i \leq G_i(x_{[i]}) \cdot p_i \leq 2\mathcal{R}(x_{[i]}) \leq 2\mathcal{R}(x^*) \ .$$

While the last inequality follows from the optimality of x^*, the middle inequality needs a more detailed explanation, depending on the relation between i and ν:

- *Case 1: $i < \nu$.* Here, we argue that the probability $\pi(i, x_{[i]})$ for product i to be purchased out of the assortment $x_{[i]}$ is at least $\frac{1}{2}$, since

$$\pi\left(i, x_{[i]}\right) = \Pr\left[U_i \geq U_\nu\right] = \Pr\left[u_i - Z_i \geq u_\nu - Z_\nu\right] \geq \Pr\left[Z_\nu \geq Z_i\right] = \frac{1}{2} \ ,$$

where the inequality above holds since $u_i \geq u_\nu$. Consequently, $\mathcal{R}(x_{[i]}) = \pi(i, x_{[i]}) \cdot p_i \geq \frac{p_i}{2}$ whereas $G_i(x_{[i]}) = 1$, so $G_i(x_{[i]}) \cdot p_i \leq 2\mathcal{R}(x_{[i]})$.
- *Case 2: $i > \nu$*: In this case, according to the expected revenue representation (1),

$$\mathcal{R}\left(x_{[i]}\right) = G_\nu\left(x_{[i]}\right) \cdot p_\nu + G_i\left(x_{[i]}\right) \cdot (p_i - p_\nu) = G_i(x_{[i]}) \cdot p_i \ .$$

B.4. Proof of Lemma 2

To prove the first part of this claim, note that since $\tilde{u}_i = \lfloor u_i \rfloor_\delta$ for every $i \in [n]$, we have $u_i - \delta \leq \tilde{u}_i \leq u_i$, by definition of the operator $\lfloor \cdot \rfloor_\delta$. Consequently,

$$\sum_{j=1}^{i} (\tilde{u}_j - \tilde{u}_i) \cdot x_j \geq \sum_{j=1}^{i} (u_j - \delta - u_i) \cdot x_j \geq \sum_{j=1}^{i} (u_j - u_i) \cdot x_j - i\delta \ ,$$

where the last inequality holds since $\sum_{j=1}^{i} x_j \leq i$. Similarly, to derive the second part, we observe that

$$\sum_{j=1}^{i} (\tilde{u}_j - \tilde{u}_i) \cdot x_j \leq \sum_{j=1}^{i} (u_j - u_i + \delta) \cdot x_j \leq \sum_{j=1}^{i} (u_j - u_i) \cdot x_j + i\delta \ .$$

B.5. Proof of Theorem 4

We first establish a relationship between the objective values of the dynamic programs $\mathcal{C}_{\mathcal{P}_\epsilon, \mathcal{U}_\epsilon}$ and $\mathcal{C}_{\mathcal{P}_\epsilon}$.

Lemma EC.5. *$(1-\epsilon) \cdot \mathcal{C}_{\mathcal{P}_\epsilon}(x) \leq \mathcal{C}_{\mathcal{P}_\epsilon, \mathcal{U}_\epsilon}(x) \leq (1+\epsilon) \cdot \mathcal{C}_{\mathcal{P}_\epsilon}(x)$, for any assortment x.*

The proof appears in Appendix B.6. Combining several of the claims that have already been utilized for proving Theorem 3, as well as the relation between $\mathcal{C}_{\mathcal{P}_\epsilon, \mathcal{U}_\epsilon}$ and $\mathcal{C}_{\mathcal{P}_\epsilon}$, we obtain

$$\begin{aligned}
\mathcal{R}(\bar{x}) \ &\geq \ \mathcal{C}_{\mathcal{P}_\epsilon}(\bar{x}) \ \geq \ (1-\epsilon) \cdot \mathcal{C}_{\mathcal{P}_\epsilon, \mathcal{U}_\epsilon}(\bar{x}) \ \geq \ (1-\epsilon) \cdot \mathcal{C}_{\mathcal{P}_\epsilon, \mathcal{U}_\epsilon}(\tilde{x}) \\
&\geq \ (1-\epsilon)^2 \cdot \mathcal{C}_{\mathcal{P}_\epsilon}(\tilde{x}) \ \geq \ (1-\epsilon)^2 \cdot \mathcal{C}_{\mathcal{P}_\epsilon}(x^*) \ \geq \ (1-\epsilon)^3 \cdot \mathcal{R}(x^*) \ .
\end{aligned}$$

Here, the first and sixth inequalities follow Claims EC.4 and EC.3. The second and fourth inequalities proceed from Lemma EC.5 (in both directions). The third and the fifth inequalities are implied by the optimality of $\bar{x}$ and $\tilde{x}$ for $\mathcal{C}_{\mathcal{P}_\epsilon, \mathcal{U}_\epsilon}$ and $\mathcal{C}_{\mathcal{P}_\epsilon}$, respectively.

B.6. Proof of Lemma EC.5

By comparing the explicit expression (7) and (10) for the objective values of $\mathcal{C}_{\mathcal{P}_\epsilon}$ and $\mathcal{C}_{\mathcal{P}_\epsilon, \mathcal{U}_\epsilon}$, respectively, it suffices to show that $(1-\epsilon) \cdot G_i(x) \leq \tilde{G}_i(x) \leq (1+\epsilon) \cdot G_i(x)$. To this end, we first observe that

$$\begin{aligned}
\tilde{G}_i(x) \ &= \ \frac{\exp(-\sum_{j=1}^{i} (\tilde{u}_j - \tilde{u}_i) \cdot x_j)}{X(i)} \\
&\geq \ \frac{\exp(-\sum_{j=1}^{i} (u_j - u_i) \cdot x_j - i\delta)}{X(i)} \\
&\geq e^{-n\delta} \cdot G_i(x) \\
&\geq (1-\epsilon) \cdot G_i(x) \ ,
\end{aligned}$$

where the first inequality follows from Lemma 2 and the last inequality is obtained by noting that $\delta = \frac{\epsilon}{2n}$, and therefore $e^{-n\delta} = e^{-\epsilon/2} \geq 1 - \epsilon$. Similarly, we have

$$\begin{aligned}\tilde{G}_i(x) &\leq \frac{\exp(-\sum_{j=1}^{i}(u_j - u_i)\cdot x_j + i\delta)}{X(i)} \\ &\leq e^{n\delta}\cdot G_i(x) \\ &\leq (1+\epsilon)\cdot G_i(x) \ .\end{aligned}$$

B.7. Proof of Lemma 3

We first show that the desired upper bound on $\tilde{U}^{\Sigma}$ is an immediate consequence of the additional termination condition, since for any profitable state

$$\begin{aligned}e^{-k_\ell\delta - \Delta} &\leq \tilde{G}\left(i, N, \tilde{U}^{\Sigma}\right) \\ &= \frac{\exp(-\tilde{U}^{\Sigma} - N\cdot(\tilde{u}_{i-1} - \tilde{u}_i))}{N+1} \\ &\leq e^{-\tilde{U}^{\Sigma}} \ .\end{aligned}$$

Now, in order to derive a lower bound on $\tilde{U}^{\Sigma}$, since $(i, N, \tilde{U}^{\Sigma}, \tilde{P}^{\mathrm{avg}})$ is a profitable state, there is a feasible sequence of transitions from $\mathcal{S}_f = (f, N_{f-1}, \tilde{U}^{\Sigma}_{f-1}, \tilde{P}^{\mathrm{avg}}_{f-1})$ to this state. Hence, $\tilde{U}^{\Sigma} \geq \tilde{U}^{\Sigma}_{f-1} + N_{f-1}\cdot(\tilde{u}_{f-1} - \tilde{u}_f)$, and we proceed by showing that $\tilde{U}^{\Sigma}_{f-1} + N_{f-1}\cdot(\tilde{u}_{f-1} - \tilde{u}_f) \geq k_\ell\delta - \Delta$. For this purpose, the important observation is that the decisions regarding products $1,\ldots,f-1$ that are captured by the initial state $\mathcal{S}_f$, i.e., none of these products are picked except for the no-purchase option when $\nu \leq f-1$, are identical to those of the assortment $x_{[f]}$, that stocks only f and ν. Therefore, $\tilde{G}(f, N_{f-1}, \tilde{U}^{\Sigma}_{f-1}) = \tilde{G}_f(x_{[f]})$. Recalling that product ℓ was chosen to be the one that maximizes $\tilde{G}_\ell(x_{[\ell]})\cdot p_\ell$, it follows that $\tilde{G}_\ell(x_{[\ell]})\cdot p_\ell \geq \tilde{G}_f(x_{[f]})\cdot p_f$. Using the equivalent representation $\tilde{G}_i(x_{[i]}) = \frac{1}{d_i}\cdot e^{-k_i\delta}$, the last inequality can be written as $\frac{p_\ell}{d_\ell}\cdot e^{-k_\ell\delta} \geq \frac{p_f}{d_f}\cdot e^{-k_f\delta}$. By combining the latter bound and the observation that $\tilde{G}(f, N_{f-1}, \tilde{U}^{\Sigma}_{f-1}) = \tilde{G}_f(x_{[f]})$, we get

$$\begin{aligned}\frac{\exp(-\tilde{U}^{\Sigma}_{f-1} - N_{f-1}\cdot(\tilde{u}_{f-1} - \tilde{u}_f))}{N_{f-1}+1} &= \tilde{G}\left(f, N_{f-1}, \tilde{U}^{\Sigma}_{f-1}\right) \\ &= \tilde{G}_f\left(x_{[f]}\right) \\ &= \frac{1}{d_f}\cdot e^{-k_f\delta} \\ &\leq \frac{1}{d_\ell}\cdot\frac{p_\ell}{p_f}\cdot e^{-k_\ell\delta} \\ &\leq \frac{p_{\max}}{p_{\min}}\cdot e^{-k_\ell\delta} \ .\end{aligned}$$

The above inequality can be rearranged to show that

$$\tilde{U}^{\Sigma}_{f-1} + N_{f-1}\cdot(\tilde{u}_{f-1} - \tilde{u}_f) \geq k_\ell\delta - \ln\left((N_{f-1}+1)\cdot\frac{p_{\max}}{p_{\min}}\right) \geq k_\ell\delta - \Delta \ .$$

B.8. Proof of Lemma 4

As any feasible assortment for $\mathcal{C}^{\mathcal{S}_f}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}$ has precisely the same objective value when evaluated under $\mathcal{C}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}$, it follows that $\mathcal{C}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}(\hat{x}) = \mathcal{C}^{\mathcal{S}_f}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}(\hat{x})$. As a result, to establish the desired claim, we proceed by proving that $\mathcal{C}^{\mathcal{S}_f}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}$ admits a feasible assortment whose objective value is at least $(1-\epsilon)\cdot\mathcal{C}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}(\bar{x})$.

To this end, recalling that $f \neq \nu$ is the minimum-index product that is picked by $\bar{x}$, the latter assortment can be viewed as starting from the initial state $\mathcal{S}_f$. However, $\bar{x}$ may not form a feasible assortment for $\mathcal{C}^{\mathcal{S}_f}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}$, as the sequence of states traversed by following the decisions of $\bar{x}$ may include non-profitable states. Nevertheless, since $\tilde{G}_f(\bar{x}) \geq \cdots \geq \tilde{G}_n(\bar{x})$, these non-profitable states are necessarily a suffix of the previously mentioned sequence. Thus, there exists an index $m > f$ for which setting the value of $\bar{x}_m, \ldots, \bar{x}_n$ to zero (i.e., not picking any of the products $m, \ldots, n$) results in a $\mathcal{C}^{\mathcal{S}_f}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}$-feasible assortment. By (10), the objective value of the latter is $\mathcal{C}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}(\bar{x}) - \sum_{i=m}^n \tilde{G}_i(\bar{x}) \cdot (p_i - \mathrm{avg}_{i-1}(\bar{x})) \cdot \bar{x}_i$. However, due to the definition of profitable states, $\tilde{G}_i(\bar{x}) < e^{-k_\ell \delta - \Delta}$ for every $m \leq i \leq n$, and therefore

$$\begin{aligned}
\sum_{i=m}^n \tilde{G}_i(\bar{x}) \cdot \big(p_i - \mathrm{avg}_{i-1}(\bar{x})\big) \cdot \bar{x}_i &\leq n \cdot e^{-k_\ell \delta - \Delta} \cdot p_{\max} \\
&= \frac{\epsilon}{4} \cdot e^{-k_\ell \delta} \cdot p_{\min} && \text{(EC.1)} \\
&\leq \frac{\epsilon}{2 d_\ell} \cdot e^{-k_\ell \delta} \cdot p_\ell \\
&= \frac{\epsilon}{2} \cdot \tilde{G}_\ell(x_{[\ell]}) \cdot p_\ell && \text{(EC.2)} \\
&\leq \epsilon \cdot \mathcal{C}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}(x_{[\ell]}) && \text{(EC.3)} \\
&\leq \epsilon \cdot \mathcal{C}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}(\bar{x}) \ . && \text{(EC.4)}
\end{aligned}$$

Here, equality (EC.1) holds since $\Delta = \ln(\frac{4n}{\epsilon} \cdot \frac{p_{\max}}{p_{\min}})$, and equality (EC.2) follows from the equivalent representation $\tilde{G}_\ell(x_{[\ell]}) = \frac{1}{d_\ell} \cdot e^{-k_\ell \delta}$. Then, inequality (EC.3) is obtained by observing that $\tilde{G}_\ell(x_{[\ell]}) \cdot p_\ell \leq 2 \cdot \mathcal{C}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}(x_{[\ell]})$. To see this, based on representation (10), when $\ell > \nu$ we have $\mathcal{C}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}(x_{[\ell]}) = \tilde{G}_\ell(x_{[\ell]}) \cdot p_\ell$. In the opposite scenario, where $\ell < \nu$, we have $\mathcal{C}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}(x_{[\ell]}) = (\tilde{G}_\ell(x_{[\ell]}) - \tilde{G}_\nu(x_{[\ell]})) \cdot p_\ell \geq \frac{1}{2} \cdot \tilde{G}_\ell(x_{[\ell]}) \cdot p_\ell$, since $\tilde{G}_\ell(x_{[\ell]}) = 1$ and $\tilde{G}_\nu(x_{[\ell]}) = \frac{1}{2} e^{-(\tilde{u}_\ell - \tilde{u}_\nu)} \geq \frac{1}{2}$ in this case. Finally, inequality (EC.4) is due to $\bar{x}$ being an optimal assortment for $\mathcal{C}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}$.

Appendix C: Additional Details on Case Studies

C.1. Estimation methods

Implementation. Since the MLE problem under the Exponomial model is rather cumbersome due to the intricate nature of the expressions for its purchase probabilities, we employ two different software packages to solve this MLE problem in order to understand the best approach for fitting the Exponomial model to historical sales data. More specifically, we utilize MATLAB and Tensorflow implementations as follows:

- *Packaged solver implementation:* Here, the MLE procedure is implemented using MATLAB's constrained nonlinear solver *fmincon*. The fmincon algorithm is based on second order methods that exploit local Hessian information.

- *Stochastic gradient-descent:* In the modern practice of machine learning, it is common to employ online estimation algorithms, where each gradient step is computed using a small (randomly generated) batch of data. In this vein, we develop a Python implementation using the Tensorflow framework. This package enables various first-order stochastic gradient descent algorithms. We specify a batch size of 100 and the learning rate is tuned in the set $\{0.001, 0.0001\}$ based on the in-sample log-likelihood. We specify the Adam optimizer as our stochastic gradient-descent method.

Running times. Both MLE procedures are implemented on a standard desktop computer, with Intel Core i5 3.2GHz CPU and 32GB of RAM. For the transit app case study, using the MATLAB implementation, the average running times across the 5 folds were approximately 500 and 2,800 seconds to fit the MNL and Exponomial models, respectively. In contrast, on the same data set, the average running times of the Tensorflow implementation are 200 and 580 seconds for the MNL and Exponomial models, respectively. Clearly, the complexity of computing the choice probabilities under the Exponomial choice model incurs larger computational cost. That said, these running times are in the same order of magnitude under both choice models, and online methods allow to speed-up the estimation procedure.

C.2. Entropy and magnitude

The goal of this section is to formalize the connection between the notions of entropy and magnitude, introduced in Section 6.3, and the sparsity of rank-based choice models, i.e., the number of rankings in the support of the corresponding distribution. We begin by proposing a natural (and rather general) randomized process for generating rank-based choice models of different levels of sparsity. Next, we show that, for any assortment, the expected entropy can only increase as a function of the sparsity of the rank-based model, while the expected magnitude can only decrease with the level of sparsity.

C.2.1. Generating rank-based models: Let $\mathcal{D}$ be an arbitrary distribution over rankings on n product alternatives and the no-purchase option, i.e., permutations over $n+1$ elements. The distribution $\mathcal{D}$ induces the following random generative process for rank-based choice models where, given a desired level of sparsity $k \geq 0$, we generate an instance $\hat{R}_k$ of the rank-based model by independently sampling k rankings $\hat{\sigma}_1, \ldots, \hat{\sigma}_k$ from the distribution $\mathcal{D}$. The choice probability of all picked rankings will be identical, namely, $\hat{R}_k$ assigns each of $\hat{\sigma}_1, \ldots, \hat{\sigma}_k$ a choice probability of $1/k$. For every assortment $A \subseteq [n]$, we denote by $\mathbb{I}(i, A, \hat{\sigma}_j)$ the indicator of whether i is the most preferable alternative according to the ranking $\hat{\sigma}_j$ out of the products included in $A \cup \{0\}$. With this notation, the choice probability prescribed by $\hat{R}_k$ for each alternative $i \in A \cup \{0\}$ is given by

$$\pi\left(i, A, \hat{R}_k\right) = \frac{1}{k} \cdot \sum_{j=1}^{k} \mathbb{I}\left(i, A, \hat{\sigma}_j\right) \ .$$

We define the entropy $H(\cdot)$ and magnitude $M(\cdot)$ as scalar functions over choice probability vectors induced by assortments (that is, each assortment $A \subseteq [n]$ induces the vector $\pi(\cdot, A, \hat{R}_k)$ of dimension $|A|+1$). The entropy associated with an instance $\hat{R}_k$ of the rank-based model and an assortment $A \subseteq [n]$ is given by

$$H\left(\pi\left(\cdot, A, \hat{R}_k\right)\right) = -\sum_{i \in A} \pi\left(i, A, \hat{R}_k\right) \cdot \log\left(\pi\left(i, A, \hat{R}_k\right)\right) \ .$$

In addition, we define its magnitude as

$$M\left(\pi\left(\cdot, A, \hat{R}_k\right)\right) = \sum_{i \in A} \left(\pi\left(i, A, \hat{R}_k\right)\right)^2 \ .$$

C.2.2. Analysis: With these definitions, we are ready to show that the expected entropy and magnitude with respect to our generative model are monotone functions of the sparsity level.

LEMMA EC.6. *For every $k \geq 0$ and for every $A \subseteq [n]$, we have:*

1. $\mathbb{E}_{\hat{R}_{k+1}}[H(\pi(\cdot, A, \hat{R}_{k+1}))] \geq \mathbb{E}_{\hat{R}_k}[H(\pi(\cdot, A, \hat{R}_k))]$.
2. $\mathbb{E}_{\hat{R}_{k+1}}[M(\pi(\cdot, A, \hat{R}_{k+1}))] \leq \mathbb{E}_{\hat{R}_k}[M(\pi(\cdot, A, \hat{R}_k))]$.

Proof. We first observe that $H(\cdot)$ is a concave function and that $M(\cdot)$ is a convex function over choice probability vectors. Consequently, it is sufficient to show that $\pi(i, A, \hat{R}_{k+1}) \preceq_{cx} \pi(i, A, \hat{R}_k)$ for every given assortment $A \subseteq [n]$ and alternative $i \in A \cup \{0\}$, where we use $\preceq_{cx}$ to denote the convex stochastic ordering between random variables. Namely, $X \preceq_{cx} Y$ if and only if $\mathbb{E}[X] = \mathbb{E}[Y]$ and $\mathbb{E}[\phi(X)] \leq \mathbb{E}[\phi(Y)]$ for every convex function ϕ. We refer the reader to the excellent book by Shaked and Shanthikumar (2007, Chap. 3) for further background on stochastic orders.

For ease of notation, let $\theta = \mathbb{E}_{\hat{\sigma} \sim \mathcal{D}}[\mathbb{I}(i, A, \hat{\sigma})]$. We observe that, by definition, $\pi(i, A, \hat{R}_k)$ follows the same distribution as the random variable $B_k = \frac{1}{k} \cdot B(k, \theta)$, where $B(k, \theta)$ is a Binomial random variable with k trials and success probability θ. Similarly, $\pi(i, A, \hat{R}_{k+1})$ follows the same distribution as $B_{k+1} = \frac{1}{k+1} \cdot B(k+1, \theta)$. Consequently, it remains to show that $B_{k+1} \preceq_{cx} B_k$. However, this inequality is a special case of Example 3.A.29 in (Shaked and Shanthikumar 2007). $\square$

C.3. Additional plots and tables

In Table EC.1, using OLS regression, we analyze the linear relationships between the performance gaps and the entropy and magnitude metrics described in Section 6.3. The control variable $SampleSize_t$ measures the sample size of the data set for the Expedia site corresponding to each observation $t \in [\tau]$. We provide statistically significant validations of Hypotheses H1 and H2. We note that magnitude and entropy are highly correlated (-98.9%). Due to the resulting collinearity, the linear specification (5) should be omitted and it is only reported for completeness. Finally, we note that, without the restriction $z_t \neq 0$, we no longer observe statistical significance on H1 and H2. However, as explained in Section 6.3, the fitted MNL and Exponomial models produce nearly identical choice probabilities for the no-purchase option which is the most frequently chosen alternative – specifically, the variance of the performance gaps on the different Expedia sites is smaller than 0.003 for the sample $\{t \in [\tau] : z_t = 0\}$ and exceeds 0.21 for the sample $\{t \in [\tau] : z_t \neq 0\}$.

Table EC.1 Relating the performance gaps to the entropy and magnitude of choice events $t \in [\tau]$ for which $z_t \neq 0$.

	$PerformanceGap_t$				
	(1)	(2)	(3)	(4)	(5)
$Intercept$	0.633***	0.711***	-1.137***	-0.938***	-4.328***
	(0.018)	(0.021)	(0.038)	(0.046)	(0.360)
$Entropy_t$	-0.603***	-0.565***			-1.738***
	(0.018)	(0.021)			(0.118)
$Magnitude_t$			1.746***	1.625***	-3.626***
			(0.057)	(0.055)	(0.361)
$SampleSize_t$		-2.83e-06***		-2.87e-06***	-3.24e-06***
		(3.71e-07)		(3.78e-07)	(3.68e-07)
R^2	0.251	0.263	0.227	0.240	0.284
Adj. R^2	0.251	0.263	0.227	0.239	0.284
Num. obs.	3482	3482	3482	3482	3482

$^{*}p < 0.05$; $^{**}p < 0.01$; $^{***}p < 0.001$.

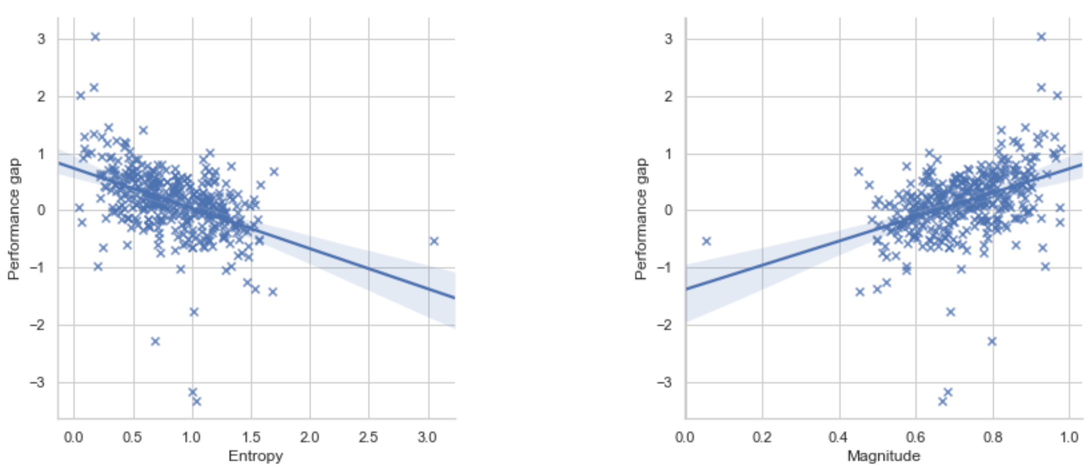


Figure EC.1 Performance gaps between the Exponomial and MNL choice models for the choice events $t \in [\tau]$, $z_t \neq 0$ on the Expedia site 2.

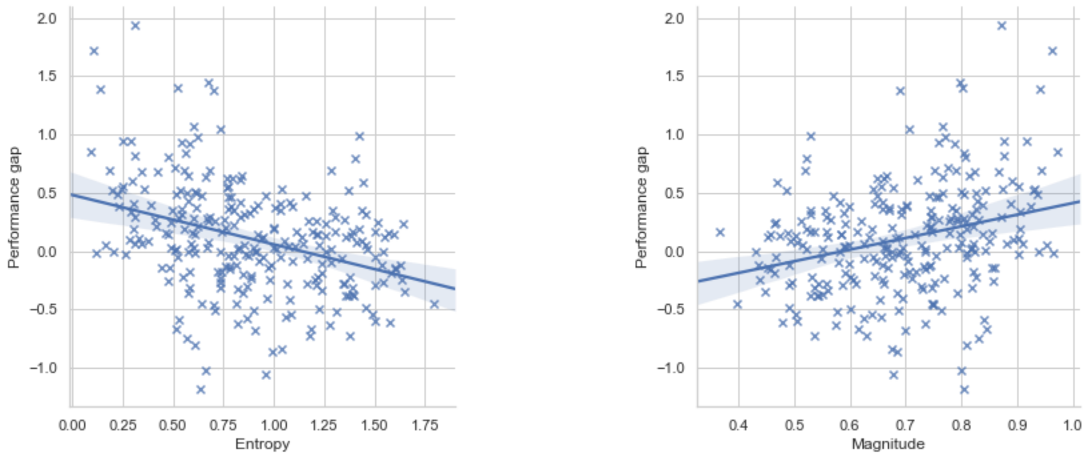


Figure EC.2 Performance gaps between the Exponomial and MNL choice models for the choice events $t \in [\tau]$, $z_t \neq 0$ on the Expedia site 3.

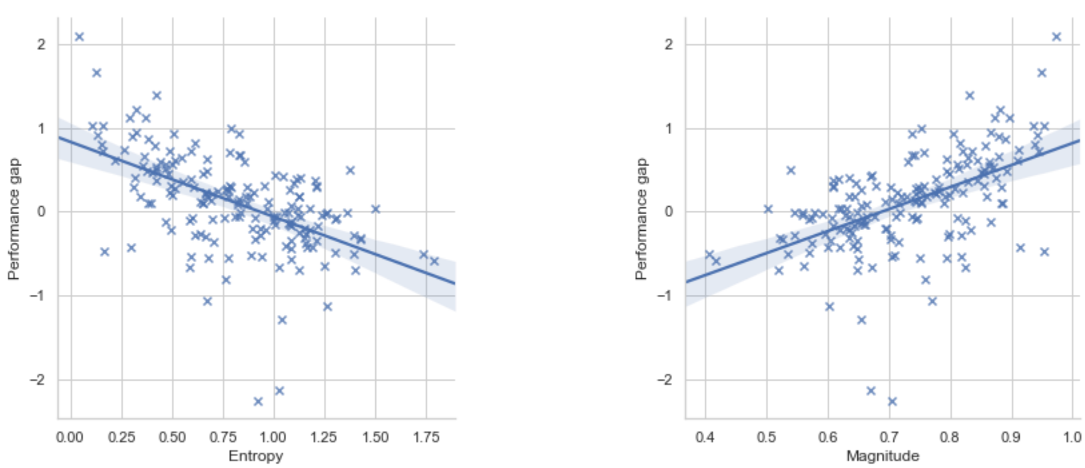


Figure EC.3 Performance gaps between the Exponomial and MNL choice models for the choice events $t \in [\tau]$, $z_t \neq 0$ on the Expedia site 4.

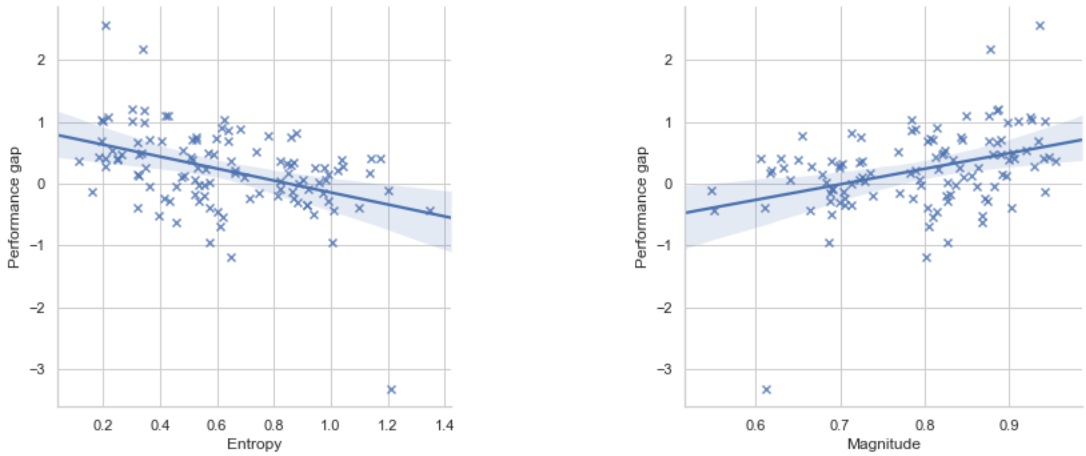


Figure EC.4 Performance gaps between the Exponomial and MNL choice models for the choice events $t \in [\tau]$, $z_t \neq 0$ on the Expedia site 5.

C.4. The structure of recommended assortments: A deeper examination

In what follows, we develop a number of hypotheses to explain the relative performance gaps of the three models; these center around the downstream effects of biases in the predicted choice probabilities. More specifically, we propose the following hypotheses to explain our observed trends:

H1 (MNL assortments): The fitted MNL models over-estimate market share, which causes the assortment prescribed by each fit to contain too few products relative to the true optimal assortment.

H2 (MC assortments): The fitted MC models occasionally underestimate market share, which causes the assortment prescribed by each fit to contain too many products relative to the true optimal assortment.

We devote the remainder of this section to presenting evidence in support of these hypotheses, which comes in the form of various plots that depict the accuracy of the fitted choice models and help visualize the structure of the recommended assortments.

Evidence in support of H1. As a starting point for our analysis, we note that the average number of products included in the assortments $S^*(\text{MNL})$, $S^*(\text{EP})$, and $S^*(\text{GT})$ are 3.16, 3.66, and 3.93, respectively. Based on this observation, the assortments recommended by the fitted MNL models generally consist of fewer products than the optimal assortment, while those recommended by the fitted Exponomial models are also sparser on average, but to a lesser degree. Figure EC.5 dives deeper into this trend by providing a fine-grained look at the structure of the recommended assortments with regards to how frequently the highest revenue products are offered. In this plot, the height of each bar at the "1" mark on the x-axis should be interpreted as the number of test instances in which the highest revenue product was included within $S^*(\text{MNL})$, $S^*(\text{EP})$, $S^*(\text{MC})$, and $S^*(\text{GT})$, whereas the bars at the "2", ..., "10" marks correspond to the second-highest up to tenth-highest revenue products. As indicated by the decreasing heights of these bars in left-to-right order, it appears that the assortments $S^*(\text{MNL})$, $S^*(\text{EP})$, and $S^*(\text{GT})$ generally include higher revenue products more frequently than lower revenue products. When this insight is combined with the fact that $S^*(\text{MNL})$ is a subset of $S^*(\text{EP})$ in 81.7% of the test instances for which $S^*(\text{MNL}) \neq S^*(\text{EP})$, it becomes very plausible that the 4% revenue improvement of the assortments recommended by the Exponomial fits can be explained by the inclusion of a few additional products in $S^*(\text{EP})$ that are left out of $S^*(\text{MNL})$. Moreover, in 72.2% of the instances tested for which $S^*(\text{MNL}) \subset S^*(\text{EP})$, we observe that $S^*(\text{EP}) \setminus S^*(\text{MNL}) \subseteq S^*(\text{GT})$, adding further support to the claim that these additional products are indeed valuable.

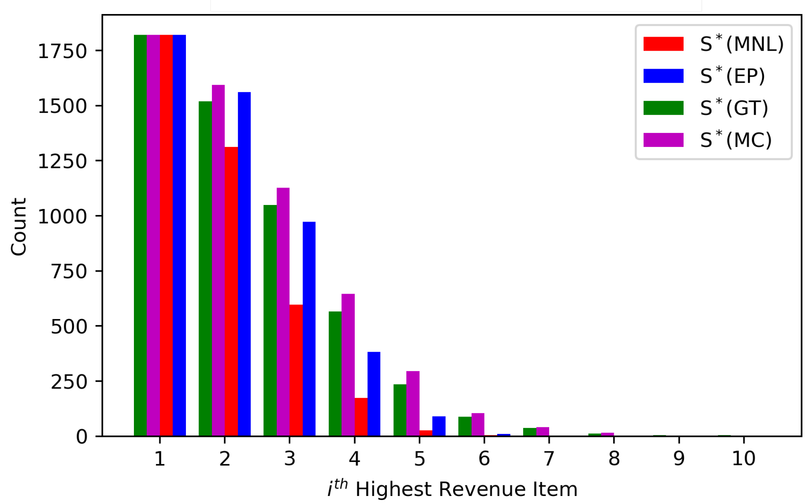


Figure EC.5 Number of instances where the recommended assortment contains the i-th highest revenue item.

As stated in H1, we believe that the phenomenon where $S^*(\text{MNL})$ contains fewer products than $S^*(\text{GT})$ may be related to the positive bias of the choice probability estimates of the fitted MNL models. The predicted market share of the assortment $S^*(\text{CM})$ is $\sum_{i \in S^*(\text{CM})} \pi_i^{\text{CM}}(S^*(\text{CM}))$, while its true market share is given by $\sum_{i \in S^*(\text{CM})} \pi_i^{\text{GT}}(S^*(\text{CM}))$. We define the error in predicted market share to simply be the difference between the predicted market share and the true market share, where a positive error corresponds to an over-estimate of the market share. The left plot of Figure EC.6 shows the distribution of these prediction errors for both the fitted MNL and Exponomial choice models over all test instances. The distribution of the MNL-related errors clearly reveals that the market share of the assortment $S^*(\text{MNL})$ is significantly over-estimated in most cases. In turn, this leads to the sparsity of the assortments $S^*(\text{MNL})$, which presume to capture a larger fraction of the market than they actually do.

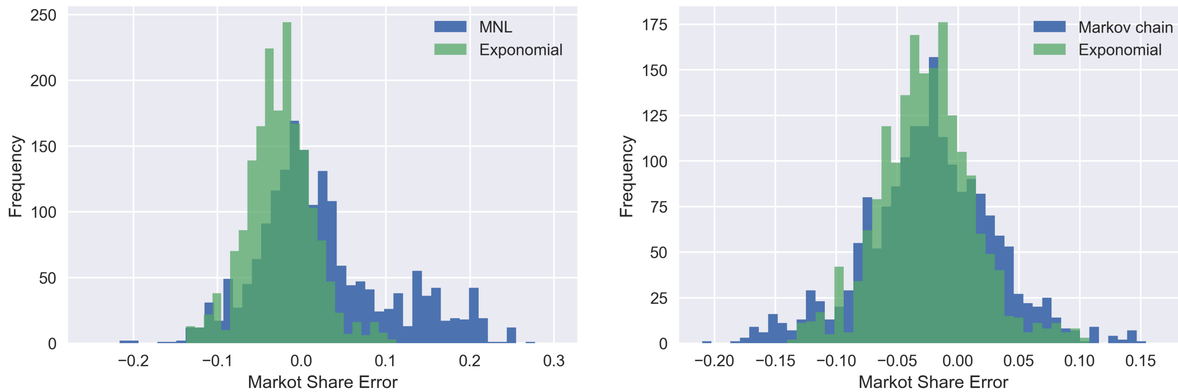


Figure EC.6 Distribution of market share prediction errors for each of the fitted models.

To clarify this last point, it is instructive to consider the following algorithmic procedure that constructs an optimal assortment $S^*(\text{MNL})$ for the MNL choice model. This procedure provides clear-cut reasoning for why over-estimating market share can lead to overly sparse recommended assortments. First, recall the following property of the expected revenue function under MNL preferences: for every assortment $S \subseteq [n]$ and product $i \in [n] \setminus S$, we have

$$\mathcal{R}(S \cup \{i\}) = \underbrace{\frac{1+\sum_{k\in S} w_k}{1+\sum_{j\in S\cup\{i\}} w_j}}_{=\alpha} \cdot \mathcal{R}(S) + \underbrace{\frac{w_i}{1+\sum_{j\in S\cup\{i\}} w_j}}_{=1-\alpha} \cdot r_i,$$

where $\alpha \in [0,1]$. The above property reveals that $\mathcal{R}(S \cup \{i\}) \geq \mathcal{R}(S)$ if and only if $r_i \geq \mathcal{R}(S)$. This insight provides a simple method to determine the lowest revenue product included in $S^*(\text{MNL})$, i.e., the revenue cut-off that determines the optimal revenue-ordered assortment. Specifically, the following approach can be employed to construct $S^*(\text{MNL})$: starting from an empty assortment, we sequentially add products in decreasing order of revenue until the expected revenue of the current assortment exceeds the revenues of all products not added so far. Hence, if the market share of $S^*(\text{MNL})$ is over-estimated, this error translates into an over-estimation of the expected revenue $\mathcal{R}(S^*(\text{MNL}))$, which in turn causes the above approach to terminate early, resulting in the recommendation of a sub-optimal assortment that leaves out what appears to be key products.

Evidence in support of H2. Again, we begin by noting that the average number of products included in $S^*(\text{MC})$ is 4.10, which exceeds the average cardinalities of 3.66 and 3.96 of the assortments $S^*(\text{EP})$ and $S^*(\text{GT})$, respectively. Additionally, Figure EC.5 provides further evidence that $S^*(\text{MC})$ generally includes too many products, as we observe that the i-th highest revenue item is always included more frequently under $S^*(\text{MC})$ than under $S^*(\text{GT})$. Finally, for the test cases in which $S^*(\text{EP}) \neq S^*(\text{MC})$, we observe that $S^*(\text{EP})$ is a subset of $S^*(\text{MC})$ in 66.7% of these instances. Moreover, for this restricted class of instances where $S^*(\text{EP}) \subset S^*(\text{MC})$, we see that

$S^*(\mathrm{MC}) \setminus S^*(\mathrm{EP}) \subseteq S^*(\mathrm{GT})$ in only 32.7% of these cases, indicating that the extra products included within $S^*(\mathrm{MC})$ might not be beneficial revenue-wise.

As summarized in H2, we believe that the root cause of the extraneous products included in $S^*(\mathrm{MC})$ are under-estimates of the offered assortments' market shares. This trend is observed in the right plot of Figure EC.6, where the market share prediction errors of the Exponomial and MC fits are juxtaposed. Naturally, since the MC fits are more competitive than the MNL fits with respect to revenue performance, a consistent pattern of under-estimated market share is not quite as salient as the reverse trend for the MNL fits. Nonetheless, considering the three distributions of market share prediction errors displayed in Figure EC.6, it is clear that the MC model is more likely to under-estimate market share than the EC model. Unfortunately, under the MC model, it is difficult to explicitly tease out the effects of such errors on the offered assortment. However, basic intuition suggests that, under substitution effects, a larger variety of products should be offered when each individual product commands a smaller market share.

Appendix D: Additional Case Studies

D.1. Transit data

This data set was collected from a Bay Area based transit app that provides a service similar in spirit to Google Maps; for confidentiality reasons, the precise identity of the app considered cannot be revealed. In this app, once start and end locations are entered, users are subsequently presented with a variety of commuting options, each coming with step-by-step instructions. We refer to each such query as a trip search. The commuting options displayed can be partitioned into three main modes of transport: transit, rideshare, and walking. The transit options consist of the many local buses and Bay Area Rapid Transit (BART), noting that a single trip search often has multiple transit options displayed, whereas the rideshare option exclusively consists of various Uber options, and the walking option is self-explanatory.

For each trip search, our data set includes various details related to the trip, the travel options displayed, and each user's interaction with the app. A snapshot of these details is given in Figure EC.7. Here, each row corresponds to a unique travel option for the given trip search, and the columns give the values of each feature. A detailed description of the available features is given in Table EC.2. Throughout our case study, we consider four Uber options: uberx, select, black, and taxi. The price for each of these options was calculated from Uber's published pricing structure (see `http://uberestimate.com/prices/San-Francisco/`), and hence does not account for surge pricing. Further, with regards to how transit options are priced, we note that there is a fixed price for each bus trip, while the price of taking BART will vary depending on the origin and destination.

search_id	mode	product	timestamp	price	travel_dist	travel_time	initial_walk	end_home	end_work	chosen
369742	rideshare	black	2016-06-02 00:42:03	16.752586	0.941579	8.033333	0.000000	0	0	0.0
369742	rideshare	select	2016-06-02 00:42:03	13.856008	0.941579	8.033333	0.000000	0	0	0.0
369742	rideshare	taxi	2016-06-02 00:42:03	6.550000	0.941579	8.033333	0.000000	0	0	0.0
369742	rideshare	uberx	2016-06-02 00:42:03	7.000000	0.941579	8.033333	0.000000	0	0	0.0
369742	transit	Church	2016-06-02 00:42:03	2.250000	0.924177	14.683333	0.583333	0	0	0.0
369742	transit	Mission	2016-06-02 00:42:03	2.250000	1.055935	15.300000	10.733333	0	0	0.0
369742	transit	Van Ness-Mission	2016-06-02 00:42:03	2.250000	1.055935	14.783333	10.733333	0	0	0.0
369742	walking	n/a	2016-06-02 00:42:03	0.000000	0.898695	18.333333	0.000000	0	0	1.0

Figure EC.7 Information available for each trip search.

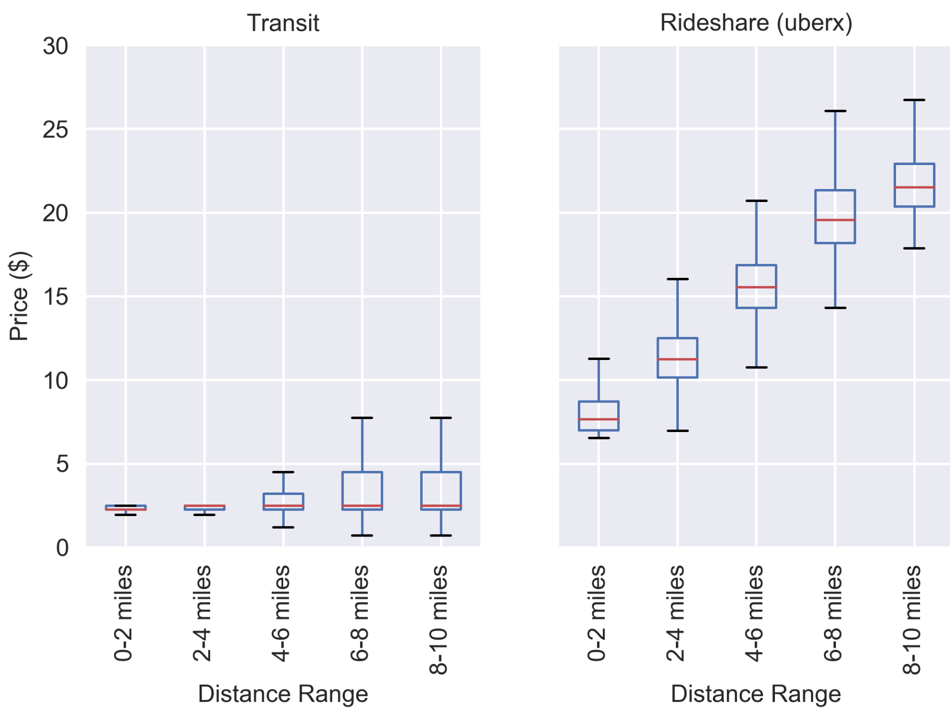
mode:	The associated mode of travel.
product:	A precise description of the travel option.
timestamp:	The date and time of the trip search.
price:	The price of the travel option.
travel_distance:	The estimated travel distance (in miles).
travel_time:	The estimated travel time (in minutes).
initial_walk:	The walking distance (in miles) to the given transit stop. This column is only relevant for transit options.
end_home:	Indicates whether the end location is close to the given user's home address.
end_work:	Indicates whether the end location is close to the given user's work address.
chosen:	The last option clicked by the user.

Table EC.2 Description of available data for each travel option.

Data preprocessing. Even though our data set includes trips that take place all over the Bay Area, we restrict the set of searches considered to those whose start and end locations are both within San Francisco County. This choice has been made in order to ensure uniformity in the trips being considered. Furthermore, we focus our attention on searches where the user clicks on at least one of the travel options displayed. After applying these filters to the set of all available trip searches, we are left with $\tau = 92,621$ searches.

Price and travel time. We conclude this summary by highlighting the trade-off between price and time that ensues when commuters can choose from a variety of travel modes. Figure EC.8 illustrates this trade-off by showing how the price and predicted travel time change as a function of the trip distance. More specifically, Figure EC.8a shows how the predicted travel time for each mode changes across trips of varying distances, and Figure EC.8b shows how the prices of the transit and uberx options vary with travel distance. Clearly, walking is always free and hence is not included in the latter plot. Each tick on the x-axis represents all trips within the specified two mile window. For example, the boxplots in Figure EC.8b at the "2-4 miles" mark show the five summary statistics of the distribution for the predicted travel time of each mode. With experimentation, we found that bucketing trip distances in increments of two miles provided the cleanest illustration of the underlying trends.

(a) Boxplots showing the predicted travel time versus travel distance for all three modes



(b) Boxplots showing price versus travel distance for transit and uberx travel options

Figure EC.8 An illustration of the price versus travel time trade-off between the three modes of transport.

As expected, we observe that walking is the slowest option while rideshare is the fastest. For trips that were under two miles, the travel times of the transit and uberx options only differ by a few minutes on average. However, for trips exceeding four miles, uberx is at least twice as fast as the transit options and approximately five times faster than walking, on average. While uberx is noticeably faster for longer trips, Figure EC.8b shows that it is also far pricier. Even for shorter trips between two and four miles, the uberx option can be over four times as expensive as the transit option. Interestingly, as the travel distance increases, the average price of the transit option stays fairly constant. This phenomenon is likely occurring since most transit options correspond to bus trips, which have a fixed cost of around $2.50.

Capturing such a trade-off, and understanding its effects on how commuters choose among the various travel options, is exactly what customer choice models are built for. By including mode specific price and travel time features in our parameterization of the Exponomial and MNL utility functions, we can understand the interplay between these two features in influencing commuter choices.

Model specification. We move on to describe how we cast the estimation problem of the MNL and Exponomial choice models. Each choice event $t \in [\tau]$ corresponds to a unique trip search. The offered assortment S_t corresponds to the travel options displayed across the three potential travel modes and z_t is the travel option selected by the user. We find that the commute choices are primarily explained by the trade-off between price and travel time. Other influential factors relate to the timing of the travel and the type of trip. Consequently, for travel option $j \in S_t$ with $m \in \{\text{walking}, \text{transit}, \text{rideshare}\}$, we express the deterministic utility as:

$$u_{jt} = \beta_0^m + \beta_1^m \cdot \text{Price}_{jt} + \beta_2^m \cdot \text{Time}_{jt} + \cdots + \beta_9^m \cdot \text{Work}_t + \beta_{10}^m \cdot \text{Home}_t \ ,$$

where all of these features are described in Table EC.3. Note that the coefficients $\beta_0^m, \ldots, \beta_{10}^m$ are mode dependent: the effects of these features vary across the different modes of transportation.

Price_{jt}:	The price of the travel option $j \in S_t$.
Time_{jt}:	The estimated travel time (in minutes) of the travel option $j \in S_t$.
Distance_{jt}:	The travel distance for the travel option $j \in S_t$.
InitialWalk_{jt} :	The walking distance (in miles) to the given transit stop associated with travel option $j \in S_t$. This feature is set to 0 for walking and rideshare.
Select_{jt} :	Indicates whether the travel option $j \in S_t$ is an uber select.
Black_{jt} :	Indicates whether the travel option $j \in S_t$ is an uber black.
Morning_t :	Indicates whether the time of trip search t is before noon.
Weekend_t :	Indicates whether trip search t occurs on a weekend.
Work_t :	Indicates whether the end location of trip search t is close to the user's work address.
Home_t :	Indicates whether the end location of trip search t is close to the user's home address.

Table EC.3 Travel option features.

Assumptions and further preprocessing. One challenge arising from the data collection process is that we do not know with certainty which travel mode was eventually selected by the user. Thus, for each trip search, we approximate this choice using the last travel option that was clicked on. We believe this assumption to be reasonable, since the last option clicked on will be the one whose directions were most recently viewed by the user. A simple analysis of the click data allowed us to confirm this intuition. First, for over 72% of the searches, only a single option was clicked, and hence we can be fairly confident that this was the option ultimately chosen by the user. In addition, Figure EC.9 shows the distributions of trip distances, broken down by mode, when the option clicked last is used as proxy for the option that was actually selected. These distributions indeed align with what one would expect to observe if we had access to the options that were actually selected. That is, the trip searches where the walking option is the last option clicked have a median trip distance of under a mile, which is three times smaller than the median trip distances of the other two modes. Finally, we make sure that our data set is not overly biased by random searches/clicks throughout the day. Figure EC.10 confirms this notion by showing how the volume of clicks changes over the course of 24 hours. As expected, we observe that most clicks occur during commuting times over the weekdays. Interestingly, for weekends, we only observe an afternoon peak, that can be attributed to the fact that most users do not make an early morning trip on weekends. In summary, we believe there is ample evidence in support of our assumption that the last option clicked is a suitable alternative for knowing the travel option that was actually chosen.

D.2. Retail panel data

We have been given access to US-wide household panel data taking the form of purchase receipts. The data sample is comprised of customer purchases made in three product categories (Dog food, Bathroom tissue, and Shampoo) across a combination of brick-and-mortar and online retailers from December 2015 to February 2016. Each transaction record that we have access to provides the brand of the purchased product, the price of this product, as well as the specific retail chain and US state where the purchase was made. For each of the three product categories, we fit separate choice models describing how customers choose amongst the set of brands available.

Model specification. Given a product category, each choice event $t \in [\tau]$ corresponds to a receipt item. Accordingly, the assortment S_t is the set of offered brands available at the time of this purchase, and $z_t \in S_t$ is the brand of the purchased product. For each brand $j \in S_t$, the deterministic component of the utility is expressed through a simple linear model:

$$u_{jt} = \beta_0^j + \beta_1^j \cdot \text{Price}_{jt} \ ,$$

where β_0^j is a brand fixed effect and β_1^j is a brand-specific price elasticity. Since we only observe purchases, the no-purchase option is excluded from the set of alternatives offered to the customer.

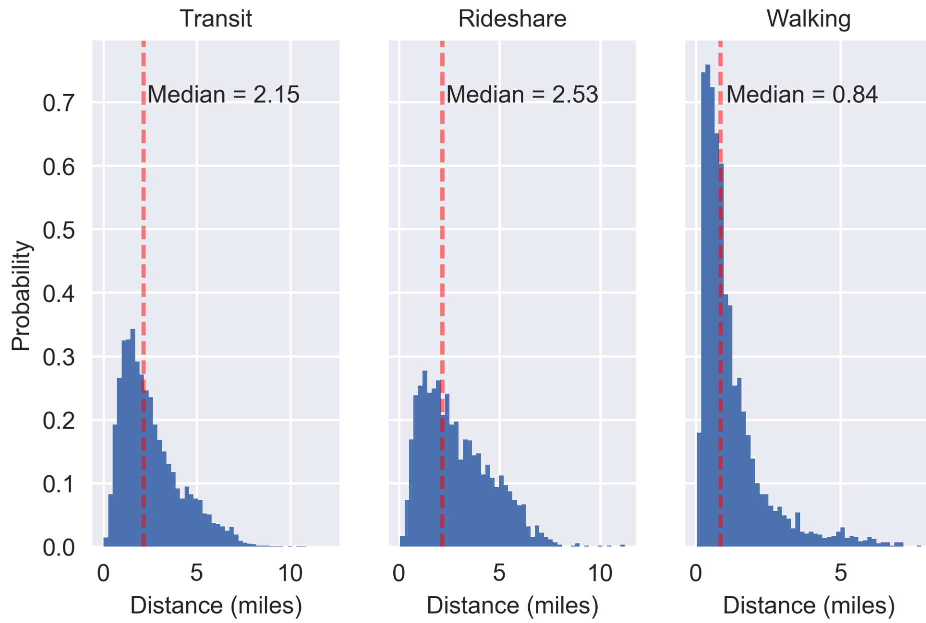


Figure EC.9 Distribution of the distance to the destination when each mode is selected.

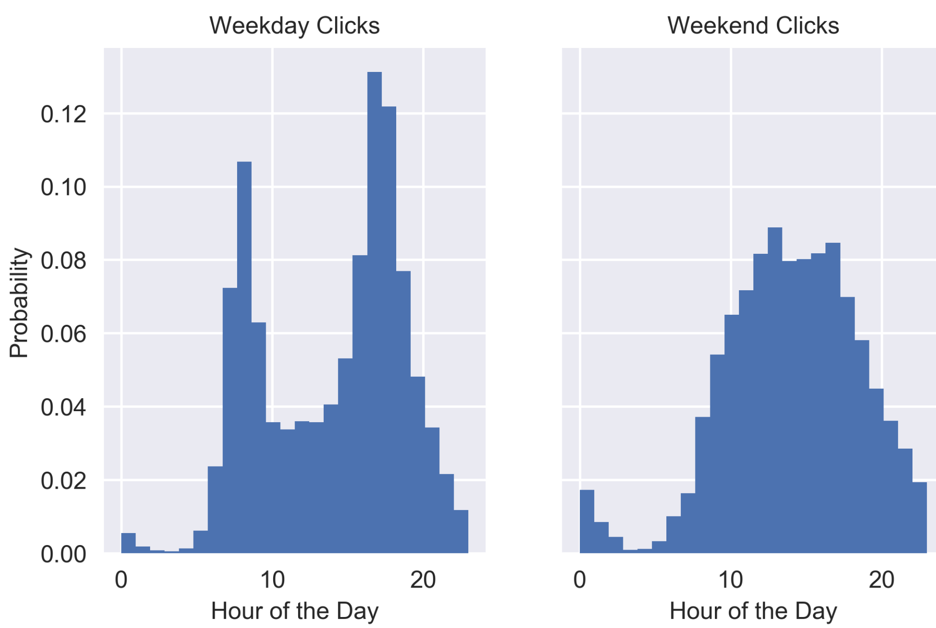


Figure EC.10 Distribution of click counts by hour of the day for weekdays and weekends.

Assumptions and data preprocessing. In contrast to the Expedia case study, this data set has a few availability limitations. For example, while each receipt specifies the brand of the purchased item, the full set of available brands S_t at the time of purchase is not directly observable. Thus, similar to other studies (Rusmevichietong and Jagabathula 2017), we make additional assumptions to approximately infer this information. Specifically, we assume that the assortment of brands available to each arriving customer at each retail chain is composed of all brands with at least one sale during the time horizon we consider. Furthermore, at the time of each transaction, we only have access to the price of the brand that was purchased and hence we are not cognizant of the posted prices of each offered brand. To overcome this issue, the price of each brand at each retail chain is estimated using a brand-specific exponential moving average model of observed purchases. That is, for every new price observation p_t, the time series model $\tilde{p}_t$ is defined by $\tilde{p}_t = \alpha \cdot p_t + (1-\alpha)\cdot \tilde{p}_{t-1}$. Through cross-validation, we pick a span parameter of 4 observations (i.e., $\alpha = 2/5$). For this inference approach to be meaningful, we filter the data by requiring that the minimal number of price observations is at least one per day on average, for each combination of brand, retailer, and state.

The assumptions detailed above are likely to lead to the formulation of an estimation problem in which the inputs are not guaranteed to match the real choice settings. For example, we assume that there is a consistent assortment of brands offered at each retail chain, when in reality, this assortment may vary due to unobservable stock-outs. Furthermore, the exponential moving average model we use to estimate prices does not incorporate store-level promotions that are likely endogenously decided based on the local levels of demand. To reiterate, since we do not have access to store-level information, we are unable to capture either of these aspects. Nonetheless, our assumptions do not bias the comparison between the MNL and the Exponomial choice models, and, as shown below, the fitted models reach satisfactory explanatory power.

D.3. Results

We carry out a 5-fold cross-validation on the transit data set. The retail data is split chronologically due to the time series nature of certain explanatory features: the training set is formed by the first 75% historical observations, while the test set is formed by the last 25% historical observations. The predictive accuracy of the MNL and Exponomial choice models are presented in Table EC.4. Focusing on the negative log-likelihood metric, we find that the Exponomial choice model outperforms the MNL choice model on the retail panel data, while the MNL model is more predictive on the transit data. The classification accuracy and rank metrics are less sensitive and show smaller gaps between the choice models.

			Classification		Rank		Neg. Log-Likelihood	
Data set	Max. $\|S_t\|$	T	MNL	Exp	MNL	Exp	MNL	Exp
Bay Area Transit	10	92621	0.62	0.62	1.66	1.66	**1.02**	1.05
Retail panel – Shampoo	39	75000	0.18	0.18	5.11	**5.01**	2.27	**2.25**
Retail panel – Dogs	54	126050	0.17	0.17	**5.89**	5.90	2.47	**2.46**
Retail panel – Tissue	45	195752	0.29	**0.32**	3.25	**3.21**	1.77	**1.76**

Table EC.4 Comparing the predictive power of the Exponomial and MNL models on additional choice datasets.

Appendix E: Computational Experiments

In this section, we turn our attention to the implementation of the FPTAS algorithm. In addition, we conduct numerical experiments to evaluate the practical computational performance of the implemented algorithm. In light of the discretization ideas developed in Sections 4.1 and 4.2, a straightforward implementation of our algorithm would generally be inefficient. That said, these ideas were devised in order to guarantee a polynomial running time for any possible instance, rather than to directly provide a practical implementation of our approach. To this end, in Section E.1 we describe a suitable adaptation of our dynamic program that prunes unnecessary states, ultimately leading to a scalable implementation of the underlying recursive approach. Through extensive experiments, we show in Section E.2 that this implementation is indeed efficient on reasonably sized instances, while still computing near-optimal assortments.

E.1. Implementation and experimental setting

Two-pass approach. Given the required accuracy level of the discretized average price and sum-of-utility-differences parameters, $\tilde{U}^{\Sigma}$ and $\tilde{P}^{\text{avg}}$, it is computationally prohibitive to enumerate all possible combinations of these parameters over the entire set $\mathcal{U}_\epsilon \times \mathcal{P}_\epsilon$. However, beyond limiting attention to profitable states, as explained in Section 5.2, it turns out that the vast majority of these states are redundant for the purpose of computing the optimal dynamic programming decisions in practice. Indeed, when one iteratively employs our recursive equations (8) and (9), starting from the initial state $\mathcal{S}_f$ and traversing only profitable states, the actual range of parameters $(\tilde{U}^{\Sigma}, \tilde{P}^{\text{avg}})$ generated along the way represents only a small fraction of all possible combinations.

With this practical observation, the implementation of our FPTAS relies on a two-pass approach. In the first pass, we tightly construct the computation tree, generating all states of interest. Namely, we start from the root node $\mathcal{S}_f$, and apply the recursive equations (8) and (9) to identify all profitable states $(i, N, \tilde{U}^{\Sigma}, \tilde{P}^{\text{avg}})$ that would ever be reached throughout the overall computation. In the second pass, we compute the function value $\mathcal{C}^{\mathcal{S}_f}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}(i, N, \tilde{U}^{\Sigma}, \tilde{P}^{\text{avg}})$ associated with each state via backward induction. In our experiments, we observe that this enumeration approach, along with the preprocessing and termination criteria developed in Section 5, significantly improve the running time by reducing the state space size.

Discretization parameters. The discretization methods developed in Sections 4.1 and 4.2 make use of the required accuracy level ϵ through the parameters $\delta = \frac{\epsilon}{2n}$ and $\theta = \frac{\epsilon}{2n^2}$. Instead, in order to avoid dealing with very small values of θ that would incur lengthy running times, our implementation directly controls the accuracy of the dynamic programming formulation by independently fixing θ, which is varied in the set $\Theta = \{2^{-1}, 2^{-2}, \dots, 2^{-7}\}$. For a given value of $\theta \in \Theta$, we define $\delta = n\theta$. It is worth noting that, while this alteration ensures that the resulting algorithm can be efficiently implemented in practice, the precise theoretical guarantee established in Section 5 may no longer hold.

Generative model. We vary the underlying number of products n over the set $\{10, 20, 30\}$, with respect to which a capacity constraint of picking at most $C = \lceil \gamma n \rceil$ products is imposed, where $\gamma \in \{0.3, 0.7, 1.0\}$. As explained in Section 8, our FPTAS can easily be adapted to capture both cardinality and capacity constraints. Note that the case $\gamma = 1.0$ corresponds to the unconstrained assortment optimization problem. In addition, the prices $p_1, \dots, p_n$ are generated through i.i.d. samples from a log-normal distribution with a location parameter $\mu = 0$ and a scale parameter $\sigma \in \{0.2, 0.5\}$. The utilities $u_1, \dots, u_n$ are generated by ordering n i.i.d. samples from the normal distribution $N(1, 1)$. Finally, similar to our conventional notation, the no-purchase option is represented by an auxiliary product $\nu \in [n]$ with utility $u_\nu = 0$ and price $p_\nu = 0$.

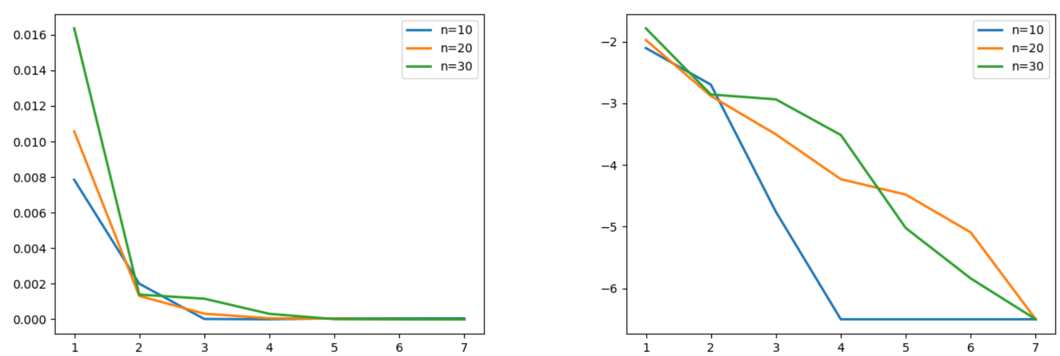
Additional details. All experiments were executed on a standard desktop computer, with an Intel Xeon 3.2GHz CPU and 32GB of RAM. Our algorithm was implemented using the Julia programming language.

E.2. Numerical results

Computation of optimality gap. Ideally, we would have been able to compute the optimality gap for each instance, defined as the percentage error $1 - \mathcal{R}(x_{\mathrm{DP}})/\mathcal{R}(x^*)$ of the dynamic programming solution x_{DP} with respect to the optimal assortment x^*. For $n = 10$ products, the optimal assortment can be obtained in less than a second through exhaustive enumeration over all subsets of at most C products. However, this brute-force approach is prohibitive for large values of n; for example, a running time that exceeds 1000 seconds is incurred when $n = 20$. Therefore, for large n values, we use a different benchmark to estimate the corresponding optimality gaps. Specifically, we define x_{b} as the assortment obtained by our dynamic program for the smallest tested value of the discretization parameter $\theta = 2^{-7}$, for which the accuracy of the FPTAS is the highest. We then estimate the optimality gap through the expression $1 - \mathcal{R}(x_{\mathrm{DP}})/\mathcal{R}(x_{\mathrm{b}})$. By comparing the benchmark x_{b} to the optimal assortment x^* obtained through enumeration when $n = 10$, we discovered that the resulting gaps are about 0.1% on average. Assuming that the accuracy of our algorithm is not dramatically affected by increasing the number of products n from 10 to 20 and 30, this finding suggests that our benchmark solution x_{b} is reasonable.

Results. In Table EC.5, we report the average and maximum optimality gaps of our algorithm against the benchmark described above. For each configuration, these metrics were computed over 50 randomly generated instances, unless the average running time exceeds 500 seconds, in which case we only generate 20 instances. In this table, the maximum optimality gap for each parameter combination is given in parentheses. In addition, σ is the scale parameter used to generate the prices, while γ is utilized to define the capacity $C = \lceil \gamma n \rceil$. Our numerical results suggest that the average optimality gaps are negligible even when using very large discretization parameters. For example, the average optimality gap is below 0.2% when $\theta = 0.25$ and $n = 10$. This observation can be contrasted with our worst-case theoretical analysis, which requires one to set the discretization parameter at $\theta = 10^{-6}$ in order to guarantee an accuracy level of $\epsilon = 0.002$. However, our approximation is sufficient for purposes of comparing different stocking decisions and obtaining near-optimal assortments. Indeed, as suggested by Figure EC.11, the convergence rate of the algorithm's error is roughly exponential in θ. It is worth noting that the different tables and figures corresponding to $n \in \{10, 20, 30\}$ are not directly comparable, since the benchmark used to normalize the optimality gap varies across the different settings.

Lastly, in Figure EC.12, we visualize the "convergence" of the recommended assortments to an optimal assortment by plotting the average cardinality of the set differences $S_{\mathrm{DP}} \setminus S^*$ and $S^* \setminus S_{\mathrm{DP}}$ as a function of θ, where S_{DP} is the assortment computed by our FPTAS and S^* is a fixed optimal assortment. Interestingly, by comparing $|S_{\mathrm{DP}} \setminus S^*|$ and $|S^* \setminus S_{\mathrm{DP}}|$, we observe that the recommended assortment S_{DP} tends to contain considerably more products in comparison to the optimal assortment S^*. However, for small values of θ, it is clear that the differences between the assortments are marginal, e.g., when $\theta = 2^{-7}$, the recommended assortment is optimal for nearly 90% of all instances.



(a) $\text{gap} = f(-\log_2 \theta)$ (b) $\log_{10} \text{gap} = f(-\log_2 \theta)$

Figure EC.11 Average optimality gap as a function of θ (logarithmic scale).

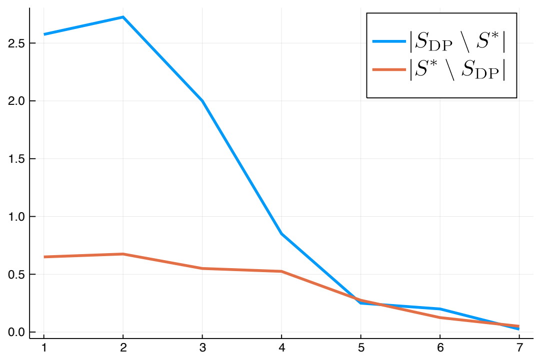
On the other hand, the maximum optimality gaps over all instances, reported in Table EC.5, are non-negligible when $\theta \in \{2^{-1}, 2^{-2}, 2^{-3}\}$. Interestingly, the optimality gaps decrease as the capacity C is increased, and the optimality gaps are the smallest in the unconstrained setting, when $\gamma = 1.0$. This observation suggests that the cardinality-constrained problem might be more challenging than its unconstrained counterpart from a computational perspective. In addition, we observe that instances with 10 products show larger optimality gaps than those with 20 and 30 products, even for small values of θ. This phenomenon could be an incidental effect of the chosen generative model,

γ	σ	n	$-\log_2\theta$ 1	2	3	4	5	6	7
0.3	0.2	10	0.7% (14.1%)	0.2% (2.8%)	0.1% (2.8%)	0.1% (2.8%)	0.1% (2.8%)	0.1% (2.8%)	0.1% (2.8%)
0.3	0.2	20	0.9% (6.8%)	0.4% (2.8%)	$< 10^{-2}$ (0.2%)	$< 10^{-2}$ (0.2%)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)
0.3	0.2	30	1.1% (7.4%)	0.5% (7%)	$< 10^{-2}$ (0.7%)	$< 10^{-2}$ (0.6%)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)
0.3	0.5	10	2.2% (15.2%)	0.9% (10.1%)	0.4% (11.3%)	0.3% (12.2%)	0.5% (12.2%)	0.5% (12.2%)	0.5% (12.2%)
0.3	0.5	20	0.3% (2.1%)	$< 10^{-2}$ (1.2%)	$< 10^{-2}$ (1.2%)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)
0.3	0.5	30	1.1% (21.9%)	0.2% (4%)	$< 10^{-2}$ (1%)	$< 10^{-2}$ (0.3%)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)
0.7	0.2	10	1.1% (10.3%)	0.4% (8.1%)	0.2% (4.8%)	$< 10^{-2}$ (2.4%)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)
0.7	0.2	20	0.7% (4.9%)	0.1% (2%)	$< 10^{-2}$ (0.2%)	$< 10^{-2}$ (0.4%)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)
0.7	0.2	30	1.4% (5.1%)	$< 10^{-2}$ (0.7%)	$< 10^{-2}$ (0.2%)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)
0.7	0.5	10	0.7% (6.6%)	0.4% (7.1%)	$< 10^{-2}$ (0.6%)	$< 10^{-2}$ (0.6%)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)
0.7	0.5	20	0.9% (10.4%)	$< 10^{-2}$ (0.1%)	$< 10^{-2}$ (1.2%)	$< 10^{-2}$ (0.1%)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)
0.7	0.5	30	0.6% (4.5%)	$< 10^{-2}$ (0.9%)	$< 10^{-2}$ (0.3%)	$< 10^{-2}$ (0.3%)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)
1.0	0.2	10	1.3% (13.7%)	0.3% (9.0%)	$< 10^{-2}$ (2.6%)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)
1.0	0.2	20	0.7% (3.2%)	0.3% (1.4%)	0.1% (0.9%)	$< 10^{-2}$ (0.1%)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)
1.0	0.2	30	0.8% (4.9%)	0.9% (3.5%)	$< 10^{-2}$ (0.1%)	$< 10^{-2}$ (0.1%)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)
1.0	0.5	10	0.8% (2.4%)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)
1.0	0.5	20	0.5% (1.7%)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)
1.0	0.5	30	0.6% (3.8%)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)	$< 10^{-2}$ ($< 10^{-2}$)

Table EC.5 Average and maximum optimality gaps.

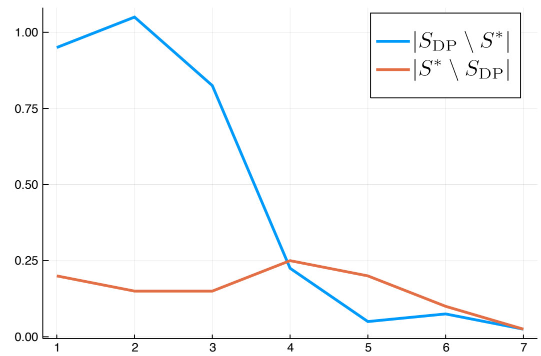
where the price and utility parameters are generated through i.i.d. samples. Due to probabilistic concentration, instances with a large number of products are more structured and result in tighter approximations.

In Table EC.6, we report the average and maximum running times in each configuration; again, the maximum running times are specified in parentheses. The algorithm is very efficient on instances



(a) $\gamma = 0.3, \;\; n = 20$

4



(b) $\gamma = 1.0, \;\; n = 20$

Figure EC.12 Average number of products in the set differences $S_{DP} \backslash S^*$ and $S^* \backslash S_{DP}$ as a function of $-\log \theta$.

comprised of at most 20 products and terminates within at most a few seconds. Furthermore, our results indicate that we are still able to compute near-optimal assortments with $n = 30$ products in less than an hour. Specifically, the largest running time of 3271 seconds is attained by an instance comprised of $n = 30$ products, with a capacity $C = \lceil 0.7 \cdot 30 \rceil = 21$, and a scale parameter $\sigma = 0.5$. As such, our approach is applicable at a moderate scale, with tens of products, which is a reasonable and relevant setting considering similar computational studies (Jagabathula 2014, Bertsimas and Misic 2019). As can only be expected, the running time increases as a function of n and C, since both parameters directly condition the number of feasible assortments. On the other hand, when σ increases, the ratio of extremal prices in a randomly generated instance tends to be larger. Indeed, since our dynamic programming approach discretizes in particular the average price parameter, the state space size and the corresponding running times are larger when $\sigma = 0.5$ in comparison to $\sigma = 0.2$.

Appendix F: FPTAS for a Capacity Constraint

In what follows, we briefly explain how to extend our approach to handle a general capacity (knapsack) constraint. By scaling, the latter corresponds to a setting where each product $i \in [n]$ consumes a w_i-portion of an overall budget, meaning that x is a feasible assortment when $\sum_{i=1}^n w_i x_i \leq 1$. Clearly, we may assume without loss of generality that every singleton assortment is feasible, i.e., $w_i \leq 1$ for every product $i \in [n]$, while the no-purchase option consumes a portion of $w_\nu = 0$.

The continuous dynamic program. Following the discussion in Sections 3-5, on top of the parameters $(i, N, \tilde{U}^\Sigma, \tilde{P}^{\mathrm{avg}})$, here the state space description is augmented with an additional continuous parameter $c \in [0, p_{\max}]$ that stands for the cumulative expected revenue generated by the stocking

γ	σ	n	$-\log_2\theta$ 1	2	3	4	5	6	7
0.3	0.2	10	$<10^{-1}$ (0.3)	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)
0.3	0.2	20	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)
0.3	0.2	30	$<10^{-1}$ (0.5)	$<10^{-1}$ (0.1)	0.3 (0.9)	0.8 (1.2)	2.5 (3.5)	5.6 (9.8)	10.4 (17)
0.3	0.5	10	$<10^{-1}$ (0.2)	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)
0.3	0.5	20	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ (1.1)	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)
0.3	0.5	30	$<10^{-1}$ (0.6)	0.2 (0.6)	0.7 (1.5)	1.9 (3.5)	4.7 (7.6)	12.7 (23)	18.6 (40.9)
0.7	0.2	10	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)
0.7	0.2	20	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)	0.1 (0.1)	0.2 (0.3)	0.3 (0.4)	0.5 (0.7)	0.6 (1.3)
0.7	0.2	30	$<10^{-1}$ (0.1)	0.3 (1.4)	1.3 (2.7)	4.3 (6.8)	13.7 (22.9)	44.2 (57.7)	225.6 (524.2)
0.7	0.5	10	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)
0.7	0.5	20	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ (0.1)	0.2 (0.3)	0.4 (0.7)	0.6 (1.2)	0.7 (1.4)	0.8 (1.5)
0.7	0.5	30	0.2 (0.3)	0.9 (3)	3 (5)	14.2 (23.9)	35.3 (65.5)	134.8 (206.7)	959.1 (3271.6)
1.0	0.2	10	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)
1.0	0.2	20	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ (0.1)	0.2 (0.3)	0.3 (0.4)	0.5 (0.6)	0.6 (0.7)
1.0	0.2	30	0.1 (0.1)	0.4 (1.9)	1.5 (2.9)	5.2 (9)	14.5 (23.2)	50.5 (72.6)	264.7 (518)
1.0	0.5	10	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ ($<10^{-1}$)
1.0	0.5	20	$<10^{-1}$ ($<10^{-1}$)	$<10^{-1}$ (0.1)	0.1 (0.2)	0.4 (0.5)	0.5 (0.7)	0.7 (0.8)	0.7 (0.8)
1.0	0.5	30	0.2 (0.3)	0.9 (3.3)	4 (6.1)	13.6 (19.6)	30.5 (49.5)	133.7 (287.3)	900 (2324.5)

Table EC.6 Average and maximum running times (in seconds).

decisions related to products $1,\dots,i-1$. Instead of the dynamic program (8), we propose an alternative formulation, where the objective is to minimize the total budget consumed by an assortment whose expected revenue is at least c^*. Thus, the new value function $\mathcal{C}^{c^*}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}(\cdot)$ is parameterized by c^*, that designates the minimal expected revenue that should be attained; we explain how c^* is determined later on. With this definition, the general recursive equation becomes:

$$\begin{aligned}
&\mathcal{C}^{c^*}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}\left(i,N,\tilde{U}^\Sigma,\tilde{P}^{\mathrm{avg}},c\right) \\
&=\min\left\{w_i+\mathcal{C}^{c^*}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}\left(i+1,N+1,\tilde{U}^\Sigma+N\cdot(\tilde{u}_{i-1}-\tilde{u}_i),\mathcal{P}_\epsilon\left(\frac{N\tilde{P}^{\mathrm{avg}}+p_i}{N+1}\right),\right.\right. \\
&\qquad\qquad\left. c+\tilde{G}\left(i,N,\tilde{U}^\Sigma\right)\cdot\left(p_i-\tilde{P}^{\mathrm{avg}}\right)\right), \\
&\qquad\left.\mathcal{C}^{c^*}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}\left(i+1,N,\tilde{U}^\Sigma+N\cdot(\tilde{u}_{i-1}-\tilde{u}_i),\tilde{P}^{\mathrm{avg}},c\right)\right\} \qquad \text{(EC.5)}
\end{aligned}$$

where $\tilde{G}(i,N,\tilde{U}^\Sigma)=\frac{\exp(-\tilde{U}^\Sigma-N\cdot(\tilde{u}_{i-1}-\tilde{u}_i))}{N+1}$. In the expression above, the first term corresponds to deciding that product i is stocked. In this case, the budget consumption is incremented by w_i, while the cumulative expected revenue is updated by the marginal contribution $\tilde{G}(i,N,\tilde{U}^\Sigma)\cdot(p_i-\tilde{P}^{\mathrm{avg}})$ of product i. The second term corresponds to the case where product i is not stocked. Here, both the budget consumption and the cumulative expected revenue are left unchanged.

Boundary cases. For the terminal case when $i=n$, the recursive equation is:

$$\mathcal{C}^{c^*}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}\left(n,N,\tilde{U}^\Sigma,\tilde{P}^{\mathrm{avg}},c\right)=\begin{cases}0, & \text{if } c\ge c^* \\ w_n, & \text{if } c+\tilde{G}(n,N,\tilde{U}^\Sigma)\cdot(p_n-\tilde{P}^{\mathrm{avg}})\ge c^* \\ \infty, & \text{otherwise}\end{cases}$$

In the first case, since the targeted expected revenue c^* has already been attained, product n is not stocked in order to minimize the total budget consumption. In the second case, product n needs to be stocked in order to meet the targeted expected revenue. In the third case, the latter constraint cannot be satisfied even when product n is stocked, and thus, we make use of ∞ to indicate that this terminal state is infeasible.

Finally, we turn our attention to the initial state of the recursion, and relate it to the overall assortment optimization problem. To this end, an optimal assortment is one that maximizes the expected revenue under the budget consumption constraint, and can therefore be cast as the problem of computing the maximum value $c^*\in[0,p_{\max}]$ for which $\mathcal{C}^{c^*}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}(1,0,0,0,0)\le 1$.

Guessing and discretization (sketch). In order to convert the continuous state space defined above into a polynomially sized one, we proceed by briefly explaining how to approximately guess c^* and how to discretize the newly added parameter c. It is worth mentioning that, unlike the rather involved arguments that were required to discretize P^{avg} and U^Σ in Sections 4.1 and 4.2, here the arguments are quite standard.

The first observation we exploit is that, at the expense of losing a $1-\epsilon$ factor in optimality, it suffices to consider the dynamic program $\mathcal{C}^{\tilde{c}}_{\mathcal{P}_\epsilon,\mathcal{U}_\epsilon}(1,0,0,0,0)$, where $(1-\epsilon)\cdot c^*\le\tilde{c}\le c^*$. To obtain such as estimate, we initially identify the product ℓ^* that maximizes $\mathcal{R}(x_{[\ell^*]})$, which is the best possible expected revenue of a single-product assortment. With this definition, it is easy to verify that $c^* \in [\mathcal{R}(x_{[\ell^*]}), n \cdot \mathcal{R}(x_{[\ell^*]})]$, and consequently, we can guess the estimate $\tilde{c}$ by exhaustively enumerating over all $O(\frac{1}{\epsilon} \cdot \log n)$ powers of $1+\epsilon$ within the interval $[\mathcal{R}(x_{[\ell^*]}), n \cdot \mathcal{R}(x_{[\ell^*]})]$.

Next, the crux in discretizing the parameter c resides in expressing the revenue contribution of every product i considered along the recursion as a multiple of $\Delta = \frac{\epsilon \tilde{c}}{n}$. More precisely, within the recursive equation (EC.5), we replace $\tilde{G}(i, N, \tilde{U}^{\Sigma}) \cdot (p_i - \tilde{P}^{\text{avg}})$ by $\lfloor \tilde{G}(i, N, \tilde{U}^{\Sigma}) \cdot (p_i - \tilde{P}^{\text{avg}}) \rfloor_{\Delta}$. Here, $\lfloor \cdot \rfloor_{\Delta}$ is an operator that rounds its argument down to the nearest integer multiple of Δ. With this modification, the potential values of the parameter c are now restricted to being (polynomially many) multiples of Δ as well. In addition, it is not difficult to verify that the cumulative additive error we incur throughout the recursive computation of $\mathcal{C}^{\tilde{c}}_{\mathcal{P}_\epsilon, \mathcal{U}_\epsilon}(1,0,0,0,0)$ is bounded by $n\Delta \leq \epsilon \tilde{c} \leq \epsilon c^*$.