Customer Choice Models versus Machine Learning: Finding Optimal Product Displays on Alibaba

Jake Feldman[1]*, Dennis J. Zhang[1]*,
Xiaofei Liu[2], Nannan Zhang[2]
1. Olin Business School, Washington University in St. Louis
2. Alibaba Group Inc.

We compare the performance of two approaches for finding the optimal set of products to display to customers landing on Alibaba's two online marketplaces, Tmall and Taobao. Both approaches were placed online simultaneously and tested on real customers for one week. The first approach we test is Alibaba's current practice. This procedure embeds thousands of product and customer features within a sophisticated machine learning algorithm that is used to estimate the purchase probabilities of each product for the customer at hand. The products with the largest expected revenue (revenue × predicted purchase probability) are then made available for purchase. The downside of this approach is that it does not incorporate customer substitution patterns; the estimates of the purchase probabilities are independent of the set of products that eventually are displayed. Our second approach uses a featurized multinomial logit (MNL) model to predict purchase probabilities for each arriving customer. In this way we use less sophisticated machinery to estimate purchase probabilities, but we employ a model that was built to capture customer purchasing behavior and, more specifically, substitution patterns. We use historical sales data to fit the MNL model and then, for each arriving customer, we solve the cardinality-constrained assortment optimization problem under the MNL model online to find the optimal set of products to display. Our experiments show that despite the lower prediction power of our MNL-based approach, it generates significantly higher revenue per visit compared to the current machine learning algorithm with the same set of features. We also conduct various heterogeneous-treatment-effect analyses to demonstrate that the current MNL approach performs best for sellers whose customers generally only make a single purchase.

Key words: choice models, product assortment, machine learning, field experiment, retail operations

1. Introduction

The assortment optimization problem has come to be one of the most well-studied problems in the field of revenue management. In this problem, a retailer seeks the revenue-maximizing set of products (or assortment) to offer each arriving customer. In its simplest form, the assortment optimization problem does not place any restrictions on the set of feasible assortments the retailer

* The first two authors contributed equally and are ranked alphabetically.

can offer. This version of the problem is often referred to as the uncapacitated assortment optimization problem. A well-studied extension of this simplest version adds a cardinality constraint, which limits the total number of products the retailer can include in any offered assortment. When each displayed product consumes the same amount of space (physical or virtual), this constraint is akin to a limit on the available shelf space or a restriction on the number of products that can be displayed on a single web page. For example, on the mobile version of Amazon, there is a section labeled "customers who bought this also bought," where at most three products can be recommended.

The central difficulty in each variant of the assortment problem is that the retailer must carefully balance the appeal of her assortment as a whole in addition to the relative appeal of the individual products that are most profitable. Adding a product to an assortment diversifies a retailer's display and thus increases her market share, but this additional product can cannibalize sales of the products that previously were a part of the assortment. The exact nature of this trade-off is dictated by the underlying customer choice model through which the retailer models customer purchasing behavior. For a fixed assortment, these models map product and customer features to the individual purchase probabilities of products included in this assortment. A variety of customer choice models have been developed in the ' ' ' 'economics and marketing literature to capture different nuances in customer purchasing behavior. Unfortunately, there is no perfect choice model, since the models that capture the most general forms of customer behavior are precisely those with an overwhelming number of parameters to estimate and whose corresponding assortment problems are difficult to solve.

The necessity to better understand this trade-off with regard to the wealth of existing choice models has given rise to two general research problems that over the last decade have guided much of the work in the field of revenue management. These two problems are summarized below.

1. **Estimation:** Can a choice model's parameters be efficiently and accurately estimated from historical data?
2. **Assortment Optimization:** Given a fully specified customer choice model, is it possible to develop efficient algorithms with provable performance guarantees for the uncapacitated assortment problem and variants thereof?

In developing answers to the questions above, the field progresses towards the ultimate goal of developing revenue management systems that use assortment optimization to guide product offering decisions. However, the recent success of machine learning methods as powerful tools for prediction calls into question the practical relevance of customer choice models and their corresponding assortment problems. *In other words, if it is indeed the case that machine learning models can* *significantly outperform choice models in terms of their ability to accurately predict customer purchasing patterns, then why would a retailer ever adopt the latter?* The answer, we find, is that accurate predictions alone are not enough to guarantee that subsequent operational decisions made from these estimates will be profitable. Of equal importance, is the sophistication with which the subsequent optimization problem captures key operational trade-offs.

We derive these insights by implementing and testing two distinct large-scale product recommender systems in collaboration with the Alibaba Group, the largest Chinese online and mobile commerce company whose Gross Merchandise Value (GMV) has surpassed US$485 billion as of 2016. More specifically, we consider a setting where Alibaba must present customized six-product assortments to inquiring customers, with the goal of maximizing revenue. The customers are presented with these personalized six-product displays after receiving discount coupons, which can be applied to any of the six offered products. Henceforth, we refer to this problem as the Alibaba Product Display Problem. The first approach that we develop and test uses the classic multinomial logit (MNL) model (Luce 1959, McFadden 1974) to capture customer preferences, and then solves a cardinality constrained assortment optimization to guide product display decisions. The second approach is Alibaba's current practice, which utilizes sophisticated machine learning methods to understand customer purchasing patterns. The implementation of both approaches unfolds in two steps that sequentially address how to estimate demand, and then how to use these estimates to identify profitable six product displays. We refer to this first step as the *estimation problem*, and the second as the *assortment problem*.

Contributions. Our foremost contribution is that we show that product recommendation systems built on the framework of the MNL model have the potential to outperform machine-learning-based approaches. We show this result by conducting a large-scale field experiment for one week in March 2018 involving more than 5 million customers. This experiment compares the performance of our MNL-based approach with Alibaba's state-of-the-art machine-learning-based approach, when both approaches use the top 25 features that are most predictive of customers' purchasing behaviors. Interestingly, we find that the fitted machine learning models produce far more accurate estimates of the purchase probabilities than the fitted MNL models, yet the MNL-based approach generates 28% higher revenue per visit. Furthermore, in September 2018, we conduct another five-day-long large-scale experiment involving more than 3 million customers, which compares our MNL-based approach with Alibaba's state-of-the-art machine-learning-based approach on all available features. In this full-feature setting, we demonstrate that our MNL-based approach generates 3.37% higher revenue per visit compared to Alibaba's current machine-learning-based recommender system. Due to data security issues, we were not given access to visit-level data from Alibaba for this second experiment so we cannot compute the accuracy of the fitted model for each approach. However, we were given the day-level aggregated results, which we used to summarize the performance of each approach.

As mentioned above, the ever-growing suite of machine learning algorithms are powerful tools for prediction; they help up us uncover and understand complex patterns in our data. However, the estimates derived from these approaches might not capture critical problem specific nuances due to the fact that they were developed as general tools. In contrast, the MNL model is simple, but it was specifically created to capture customer purchasing behavior, and more specifically, to account for substitution patterns; the phenomenon that describes the event in which a customer settles for a suitable alternative when she does not find her most preferred product available for purchase. Consequently, when this classic model is used to capture the demand for each product, we ultimately consider a more nuanced version of the assortment problem, in which substitution effects are critically accounted for.

Our second contribution comes in the form of a novel approximation scheme for a special constrained version of the assortment problem under the MNL model. Our field experiments only consider the problem of choosing optimal six product displays, however there are other operational levers that Alibaba could exploit in this discount coupon setting to increase revenues. In Appendix B, we consider two such levers, namely price and icon size. For the pricing problem, Alibaba must simultaneously decide which products to offer as well as the prices to charge for each of these offered products. For the icon size problem, we allow for Alibaba to choose the size of the product icon for each displayed product. In this setting, there is a limit on the total available screen space for all displayed products, but we do not enforce that exactly six products have to offered. Our work in this section builds off the results in Davis et al. (2013) and Feldman and Paul (2017), who both provide general schemes for solving constrained assortment problems. We present these results in Appendix B in an effort to avoid breaking up the exposition of the two approaches that we develop and test, which are the backbone of our work.

Finally, while we make substantial progress in establishing the practical underpinning for assortment optimization, there is still plenty of work to be done to cement this notion. Along these lines, an important contribution of our work is that it sheds light on new directions for work on assortment optimization that is focused on shifting research in this realm closer to the sphere of practicality. We delay a detailed description of these new problems until Section 7, since their importance is magnified once the details of our current system, along with its accompanying flaws, are understood. Nonetheless, we highlight the potential for future work early on to emphasize that even though our current MNL-based approach is quite fruitful, there are many opportunities for improvement.

Organization. The majority of the remaining content is centered around providing all of the necessary details of our field experiments. Along these lines, in Section 2, we provide a detailed overview of the discount coupon setting that we consider on Alibaba, which is the canvas for our field experiments. Sections 3 and 4 describe how we address the respective estimation and assortment problems within the two approaches that we test. Our goal in these two sections is not only to provide important implementation details, but also to highlight advantages and disadvantages inherent to each approach. This discussion helps us explain the superior performance of our MNL-based approach, and also sheds light on additional advantages of using choice models to capture customer purchasing patterns. The full details and results of our field experiments regarding the 25-feature versions of both approaches are given in Sections 5 and 6 respectively. Due to the data availability issues mentioned above, the results for the experiments comparing the full-feature versions of the two approaches is relegated to Appendix D.

1.1. Related Literature

There is an expansive collection of previous works regarding customer choice models and their accompanying assortment and estimation problems. Consequently, it is beyond the scope of this work to provide a full summary of all the past studies with related themes. Instead, since the focus of this paper is on the MNL choice model, we review past work that primarily relates to logit-based choice models, including the MNL, mixed multinomial logit (Mixed-MNL) and nested logit choice models. In doing so, we highlight the advantages and disadvantages of using each of these choice models with regard to the tractability of their corresponding assortment and estimation problems. We also include a summary of past work on the network revenue management problem, and in doing so, we illustrate that the revenue management community has implicitly considered the trade-offs between using machine learning models versus customer choice models to capture demand for many years. However, they have done so without any sort of empirical test.

The MNL choice model and mixtures thereof. The MNL choice model is perhaps the most well-studied choice model in the revenue management literature. As mentioned earlier, the MNL originally was conceived by Luce (1959) and its practical use later was most notably established by McFadden (1974), who, among other things, shows that the log-likelihood function is concave in the model parameters. To the best of our knowledge, Vulcano et al. (2012) are the first to explicitly consider estimating the parameters of an MNL choice model in a revenue management context. Instead of directly maximizing the log-likelihood, they develop an iterative expectation-maximization (EM) approach that is based on uncensoring the most preferred product of each customer. Later, Vulcano and Abdullah (2018) use a similar minorization-maximization (MM) algorithm to estimate the parameters of an MNL model from historical transaction data. They show that this newly proposed technique produces accurate estimates while being computationally superior to the previously mentioned EM approach.

The seminal works of Talluri and van Ryzin (2004) and Gallego et al. (2004) establish that the assortment optimization problem under the MNL admitted an optimal polynomial-time algorithm. These works show that the optimal assortment in this setting is a so-called revenue-ordered assortment that consists of some subset of the highest revenue products. When a cardinality constraint is added to the assortment problem, Rusmevichientong et al. (2010) provide a purely combinatorial polynomial-time algorithm, which is able to identify the optimal assortment. Building on this result, Davis et al. (2013) show that the MNL assortment problem subject to any set of totally unimodular (TU) constraints can be formulated as a concise linear program. They go on to show that a variety of realistic operational constraints can be encoded as TU constraint structures, including various forms of the aforementioned cardinality constraint.

The mixed MNL choice model segments the customer population into multiple customer types whose buying decisions are each governed by a unique MNL model. Interestingly, McFadden and Train (2000) show that this mixed MNL model is the most general choice model built on the classic framework of random utility maximization (RUM), in which arriving customers associate random utilities with each offered product and then purchase the product with the largest positive utility. Given that the mixed MNL model has the potential to capture a broad spectrum of consumer purchasing behavior, there has been much work in recent years that studies its corresponding estimation and assortment problems. For example, Subramanian et al. (2018) formulate the estimation problem as an infinite dimensional convex program and then provide a conditional gradient approach that exploits the structure of the choice probabilities to yield a local maximum of the log-likelihood. This modeling richness comes at a cost, however, as Désir et al. (2014) show that it is NP-Hard to approximate the assortment problem under the mixed MNL model within a factor of $O(n^{1-\epsilon})$ for any $\epsilon > 0$. In fact, Rusmevichientong et al. (2014) show that the problem remains NP-Hard even when the underlying population is described by only two customer types. On the positive side, Désir et al. (2014) provide an fully polynomial-time approximation scheme (FPTAS) for the assortment problem whose run time scales exponentially in the number of customer types. When the preferences of the customer types take on a special nested structure, Feldman and Topaloglu (2017) show that an FPTAS can be salvaged whose run time is polynomial in all input parameters.

The nested logit choice model. Under the nested logit choice model, the products are partitioned into nests and each customer's purchasing process unfolds in two steps: the customer first selects a nest and then makes a purchase from among the products in the chosen nest. We point the reader to Train (2009) for an excellent formal description of this model as well as an iterative maximum likelihood approach for estimating its underlying parameters. Davis et al. (2014) are the first to consider the uncapacitated assortment problem under the nested logit model. They show that the problem's hardness depends on whether the customer is allowed to leave the store without making a purchase after having selected a nest. In cases where a customer must make a purchase, the authors show that the problem admits a polynomial-time algorithm, which cleverly exploits the structure of the choice probabilities. On the other hand, the assortment problem is shown to be NP-Hard in cases where the customer can leave the store without making a purchase at either of the two steps in the purchasing process. Gallego and Topaloglu (2014) and Feldman and Topaloglu (2015) devise constant factor approximations for various constrained versions of the assortment problem under the nested logit model. Li et al. (2015) provide an exact polynomial-time algorithm for assortment optimization under the d-level nested logit model, in which nesting of the products is d-levels deep.

Network revenue management. The revenue management community has implicitly considered the trade-offs between using machine learning models versus customer choice models to capture demand for many years but without a formal empirical test. Consider, for example, the classical network revenue management problem, where the goal is to adjust the set of offered products over a selling horizon when the sale of each product consumes a combination of resources. On the one hand, there are numerous papers (Talluri and Van Ryzin 1998, 1999, Topaloglu 2009, Adelman 2007) that consider this problem under the so-called independent demand model, where the probability that product j is purchased in time period t is given by p_{jt}. On the other hand, there are many other works (Zhang and Cooper 2005, Zhang and Adelman 2009, Talluri 2010, Gallego et al. 2011) that consider the same problem when customer purchasing patterns are guided by a choice model, and so the probability that product j is purchased in time period t is $P_j(S_t)$,where S_t is the assortment of products offered in time period t. In practice, it is unclear which of these two approaches will be most lucrative. Further, the fundamental trade-off between the two approaches is exactly the one we consider when evaluating the two approach we implement for the Alibaba Display Problem. Namely, when customer behavior is governed by the independent demand model, the purchase probabilities can be accurately estimated with sophisticated machine learning methods. However, when these estimates seed the subsequent optimization problem whose solution dictates the set of products offered in each time period, this decision will once again be made without accounting substitution effects between products.

Recommender Systems We also contribute to a broad literature that studies how to design recommender systems in digital platforms (for a comprehensive review, please refer to Ricci et al. (2011)). The traditional recommender system literature often focuses on collaborative filtering; a matrix factorization technique that helps infer a customer's preference towards a product based on propensities of similar customers (Breese et al. 1998). Recently, researchers have started to consider ensemble learning methods to estimate the click-through rates or purchase probabilities of each customer on recommender platforms (Jahrer et al. 2010). Covington et al. (2016) detail YouTube's deep learning model that is used to learn each user's click-through rate, which then feeds into the videos recommendations that this user receives. Similar to the machine-learning-based approach employed by Alibaba, YouTube's recommender system relies more heavily on the estimation phase, and then solves a simple optimization problem to determine the personalized set of recommended videos.

Retailing operations. Lastly, our paper relates closely to the literature that studies assortment (Caro and Gallien 2007, Gallego et al. 2016), inventory (Caro and Gallien 2010, Cachon et al. 2018), and pricing (Ferreira et al. 2015, Papanastasiou and Savva 2016) problems in a retailing context. Caro and Gallien (2010) provide early seminal work in this stream, designing and implementing a system to help fashion retailer Zara distribute limited inventory across stores. Ferreira et al. (2015) incorporate machine learning with optimization and work with online retailer Rue La La to design a dynamic pricing system. Cachon et al. (2018) estimate the impact of inventory on sales at car dealerships and propose an inventory policy to maximize variety. Golrezaei et al. (2014) considers a dynamic assortment problem in which a retailer is allowed to personalize the assortment of products offered to each arriving customer in response to just-revealed features and the current inventory levels of each product.

2. Alibaba's Retail Setting and Product Display Problem

In this section, we begin by detailing the retail context on Alibaba where we run our field experiments. After introducing and describing this setting, we provide a general formulation of the Alibaba Product Display Problem along with a high-level overview of the two approaches that ultimately are implemented. The fundamental difference between these two approaches is the manner in which customer demand is modeled and estimated. In the first approach, which is Alibaba's current practice, machine learning models are used to estimate customer buying patterns and then a simple optimization problem is solved to choose which products to display. The second approach captures customer buying patterns through the MNL choice model, whose estimated parameters seed the assortment optimization problem that we then solve to guide product display decisions.

2.1. The Alibaba Platform

We begin by broadly discussing the two online marketplaces, Taobao.com and Tmall.com, that Alibaba has fostered to help connect third-party sellers to consumers; these are the platforms where we conduct our experiments. To help motivate our goal of maximizing revenue, we also describe how Alibaba monetizes from these two marketplaces.

Figure 1 Recommendation on Alibaba in the Public (left) and Private (right) Domains

Taobao.com is China's largest peer-to-peer retailing platform for small businesses and individual entrepreneurs. There are no commissions or listing fees on Taobao.com, and hence Alibaba monetizes its services on Taobao.com by charging fees for advertisements and seller-side helping services, such as forecasting and marketing tools. Tmall.com is China's largest third-party business-to-consumer platform for branded goods, such as Nike and Adidas. Sellers on Tmall.com are required to pay a minimum deposit when opening a store and an annual commission fee to Alibaba based on their revenue on the platform. This commission fee ranges from 0.5 to 5 percent depending on a seller's product category.[1]

With regard to Tmall.com, it is clear that Alibaba garners a larger profit when customers spend more, since Alibaba collects a small fraction of each seller's revenue for this marketplace. In the case of Taobao.com, it is also generally believed that Alibaba's profits are proportional to revenue because sellers whose revenues are largest are also those who are likely to spend more on advertising. Consequently, Alibaba primarily uses total revenue to assess the profitability of product display algorithms, even though the company does not take commission fees from sellers on Taobao.com. Hence throughout this paper, our objective is always to maximize the total revenue of each arriving customer. For the sake of brevity, we hereafter refer to Taobao.com and Tmall.com jointly as "the Alibaba platform."

[1] http://about.tmall.com/tmall/fee_schedule

2.2. The Alibaba Platform's Product Display Problem

We begin with a high-level overview of product display systems on Alibaba before focusing on the exact nature of the setting that we study. Throughout the paper, we use the terms "product display system" and "product recommendation system" synonymously. Given that the Alibaba platform essentially is a two-sided marketplace that matches customers with sellers, it is no surprise that Alibaba devotes considerable attention to developing optimal product display algorithms to ensure customers are shown products that are profitable. Broadly speaking, Alibaba's recommendation algorithms cater to two distinct settings, which we refer to as "the public domain" and "the private domain." Product recommendation algorithms applied in the public domain are applied across the entire platform and hence are not seller-specific. For example, the Tmall marketplace front page on Alibaba's mobile application (as shown in the left panel of Figure 1) is considered public domain. The product recommendation problem in this case is that of finding the optimal set of products across all sellers on the platform for each arriving customer. On the other hand, the private domain refers to pages that are specific to a particular seller. For example, the front page of Hstyle, the largest online women's apparel company on Alibaba's platform, is considered private domain (as shown in the right panel of Figure 1). Product recommendation algorithms on the private domain only promote products specific to an individual seller. We reiterate that all recommendation algorithms on Alibaba are highly personalized; if a customer lands on the front page of Tmall twice in a single day, for example, it is possible the recommended products may change as a result of this customer's interactions (clicks, searches, purchases, etc. ...) within the app between arrivals.

Our Alibaba Product Display Problem falls within the realm of private-domain product recommendation algorithms. In particular, we focus on a product recommendation problem that results when customers are given seller-specific discount coupons. Customers acquire these coupons by clicking on a coupon icon that is presented at the top of each seller's front page. Upon acquiring the coupon, customers enter a coupon sub-page that contains six displayed products, each of which can be purchased at a discount using the coupon. Alibaba chooses to display only six products since this is the largest number of products that can be displayed within a single page on a mobile device. Figure 2 shows how a customer progresses from a seller's front page to the coupon sub-page to the six displayed products. We note that this coupon feature is only available on Alibaba's mobile app, but this does not limit the scope of our experiments since the majority of Alibaba customers use the mobile app to shop. As evidence, in fiscal year 2017, the mobile GMV (i.e., revenue generated through mobile devices) was RMB 2,981 trillion (equivalent to USD 436 trillion), representing 79% of total GMV through all channels.[2]

[2] https://www.alibabagroup.com/en/news/press_pdf/p170518.pdf

Figure 2 The Process of Landing on Our Recommendation Page

Our field experiments focus exclusively on two competing approaches for finding the revenue-maximizing set of six products to make available to each customer who visits a coupon sub-page, as shown in Figure 2. As of March 2018 (just before our experiment), there were approximately 250 thousand sellers on the Alibaba platform who offered the mobile coupon discounts. On a weekly basis, these sellers witness over 25 million unique page views on their coupon sub-pages and generate over RMB 127 million (equivalent to USD 20 million) in GMV. Consequently, even small improvements to this aspect of Alibaba's recommendation systems can lead to huge gains in profit.

To help formalize our Alibaba Product Display Problem, we let $\mathcal{N} = \{1, \ldots, n\}$ be the universe of products that a particular seller potentially could offer on the coupon sub-page. Sellers on the Alibaba platform typically have between 100 to 2000 unique SKUs, all of which are in the same product category and hence can be loosely considered substitutes. We let r_j be the revenue of product $j \in \mathcal{N}$, which represents the revenue garnered from the sale of a single unit of product j. We let P_{jt} be the probability that customer t purchases product j. As indicated by its dependence on t, this purchase probability term will be uniquely determined for each arriving customer. The Alibaba Product Display Problem for customer t is given below:

$$\max_{S \subseteq \mathcal{N} : |S| = 6} \sum_{j \in S} r_j \cdot P_{jt}. \qquad \textit{(Alibaba Product Display Problem)}$$

In order to fully formulate the above problem, we must first choose a functional form for the purchase probabilities P_{jt}. We consider two alternatives, both of which parameterize the purchase probability term using a number of product and customer features. In both cases, the dependence of P_{jt} on these features is estimated from historical sales data: the estimation problem. These estimates then seed the *Alibaba Product Display Problem*, for which an efficient algorithm must be developed: the assortment problem. Along these lines, we assume throughout the paper that an "approach" for the *Alibaba Product Display Problem* includes both a procedure for deriving estimates of the purchase probabilities from historical sales data and an algorithm to find the optimal six product displays once the purchase probabilities have been estimated. Further, when discussing algorithms for "solving" the *Alibaba Product Display Problem*, we are strictly referring to developing an algorithm to solve the assortment problem.

3. The Estimation Problem

In this section, we describe the two approaches used to estimate the purchase probabilities P_{jt} that seed the *Alibaba Product Display Problem*. The first approach embeds hundreds of product and customer features within sophisticated machine learning algorithms. This approach is Alibaba's current practice for solving the estimation problem. The second approach fits featurized MNL models to the historical sales data using maximum likelihood estimation (MLE). While the latter approach can be described in full detail, we are are not able to provide the exact details of the machine-learning-based approach due to confidentiality concerns; however, we intend to provide enough details so that the advantages and drawbacks of this approach can be well understood. Further, at the conclusion of this section, we provide a case study comparing the fitting accuracy of an off-the-shelf machine learning algorithms with that of the MNL model using historical sales data from the top ten sellers (based on traffic) from Tmall.com. The intent of this case study is to show that it is not difficult to develop machine-learninh-based estimation schemes that outperform the MNL model in terms of fitting accuracy. However the essential question that we ultimately investigate in Section 6, where the results of our large-scale field experiment are revealed, is whether these gains in fitting accuracy lead to more profitable six-product offerings on the coupon sub-pages.

Available sales data and product/customer features. Before diving into the details of either approach, we first discuss the makeup of the available historical sales data used to fit the machine learning algorithms and the MNL model. This training data is composed of historical sales information from τ past customers, each of whom is shown six products. For each arriving customer t, we let $S_t \subseteq \mathcal{N}$ be the six displayed products, which the system stores as vectors of representative feature values. The product features that are used include high-dimensional static features, such as a one-hot encoding representation of product ID and seller ID, in addition to low-dimensional static features, such as product category. Dynamic product features, which are updated constantly based on customer interactions, are also included in the feature set. Examples of dynamic product features include the number of reviews and price, which are refreshed every second. Finally, we note that product features are also engineered from product descriptions and pictures. For example, there is a feature associated with the image quality of each product's icon that is displayed to customers within the app. The system also records an associated feature vector that describes the characteristics of the customer at hand. The customer-specific features include demographic information, such as age, gender, and registration time. Other customer features are descriptive of past behaviors within the app, e.g., the number of products viewed, collected, purchased, and returned in the past.[3]

Beyond the classic product/customer features described above, the system also records dynamically updated joint features of each customer and product pair. These joint features can be thought of as scores that represent estimates of the extent to which the particular product will appeal to the particular customer. These scores are computed by a large collaborative filtering system (Linden et al. 2003), which uses past purchase and click data from the given customer and other customers who are deemed to have similar preferences. Since these collaborative filtering scores depend on customer behavior within the app, they are dynamically updated so that they reflect current trends. In total, hundreds of features – numerical and categorical, static and dynamic – are available to be used within the estimation schemes.

3.1. Fitting Machine Learning Models

In what follows, we formalize the machine-learning-based approach for estimating the purchase probabilities P_{jt}. Each observation within the training data set can be described as a triple (X_{jt}, C_{jt}, Z_{jt}) corresponding to a specific arriving customer t and displayed product $j \in S_t$. The vector X_{jt} gives the features associated with the particular observation, while the output or target variables $C_{jt}, Z_{jt} \in \{0,1\}$ denote whether customer t clicked or purchased displayed product j respectively. We set $C_{jt} = 1$ if customer t clicked on product j and $C_{jt} = 0$ otherwise. Similarly, we set $Z_{jt} = 1$ if customer t purchased product j and $Z_{jt} = 0$ otherwise. Note that we must have $Z_{jt} \leq C_{jt}$ since a product cannot be purchased unless it is clicked. In total, the training data consists of $T = 6\tau$ (since each customer is shown six products) observations, which we represent as $\text{PurchaseHistory}_{ML} = \{(X_{jt}, C_{jt}, Z_{jt}) : t = 1, \ldots, T, j \in S_t\}$. We note that for this approach, each observation $(X_{jt}, C_{jt}, Z_{jt}) \in \text{PurchaseHistory}_{ML}$ does not encode the set of products that were offered alongside product j to customer t. On the one hand, this is the classic setup of supervised learning problems, which makes the task of estimating customer click and purchase probabilities amenable to the full suite of powerful machine learning tools. However, the drawback of this approach is that the estimates of the purchase probabilities are *independent* of the assortment of products displayed and hence do not account for customer substitution behaviors. Consequently,

[3] "Collecting" a product on Alibaba is analogous to adding a product to a wish list on Amazon.

the efficacy of the resulting *Alibaba Product Display Problem* in identifying profitable six-product displays could suffer due to the fact that it does not account for key operational trade-offs.

The training data is used to solve two independent estimation problems, which are then combined to form estimates of the purchase probabilities P_{jt}. First, the training data is used to derive estimates of the click probabilities $\mathbb{P}(C_{jt} = 1)$, which represent the probability that customer t will click on product j. To do so, various machine learning algorithms are employed, which are finely tuned to match the past click history described in $\text{PurchaseHistory}_{ML}$. We let the output of this estimation procedure be a function $f(X_{jt})$, which maps customer and product features to estimates of click probabilities. Along the same lines, Alibaba tries a similar collection of machine learning approaches to uncover a function $g(X_{jt})$, which produces accurate estimates of the conditional purchase probabilities $\mathbb{P}(Z_{jt} = 1 | C_{jt} = 1)$. Ultimately, Alibaba uses $P_{jt}(X_{jt}) = f(X_{jt}) \cdot g(X_{jt})$ as their estimates of the purchase probabilities, where we now explicitly express this probability as a function of the feature vector X_{jt}. It is important to note that in this setting, the estimates of the purchase probabilities are *independent* of the displayed assortment.

The current system implements various models and ensembles their predictions together for both estimation problems. These models include regularized logistic regression (Ravikumar et al. 2010), gradient-boosted decision trees (Friedman 2002), and deep learning (LeCun et al. 2015). As of the time our system is deployed (i.e., March 2018), regularized logistic regression and gradient-boosted decision trees contribute the most to the final prediction outcome due to their superior prediction performance compared to that of deep neural networks. The implementation of these machine learning algorithms is conducted offline using historical purchases from a seven-day rolling window. For example, the model on March 8, 2018, will be trained on observations from March 1, 2018, to March 7, 2018, and the model on March 9, 2018, will be trained on data from March 2, 2018, to March 8, 2018. On average, we have between 20 million and 30 million observations within these seven-day windows. It takes approximately 30 minutes to train the machine learning model and upload the result to the parameter cache server to speed up inference.

3.2. Fitting The MNL Model

In this section, we formally introduce the MNL choice model and describe how its underlying parameters are fit to historical sales data. The fitted parameters of the underlying MNL model are then used to derive estimates of the purchase probabilities that seed the *Alibaba Product Display Problem*. In contrast to the machine-learning-based approach, the MNL model is simple, but it was created with the intention to capture customer purchasing behavior and, more specifically, substitution patterns. Consequently, while the estimates produced from the fitted MNL models might not be as accurate as those produced by the machine learning based approach, the resulting *Alibaba Product Display Problem* is more sophisticated due to the fact that the purchase probabilities will be a function of the displayed assortment of products.

The MNL choice model. We begin with a description of the classic MNL choice model. The MNL choice model falls under the general RUM framework, in which arriving customers associate random utilities with the offered products and are then assumed to purchase the product with the highest positive utility. Under the MNL choice model, the random utility U_{jt} that customer t associates with product j is written as the sum of a deterministic component V_{jt} and an i.i.d. Gumbel random variable denoted as ϵ_{jt}. More formally, we have that

$$U_{jt} = V_{jt} + \epsilon_{jt}.$$

In order to incorporate product and customer features within the above utility function, one can write the deterministic component of the utility as $V_{jt} = \beta' X_{jt}$, where the vector X_{jt} denotes the values of the relevant features for customer t and product j. In this setting, we featurize the utility functions using only the top 25 product and customer features based on feature importance scores that the machine learning estimation algorithms return.

With this notation in hand, we can present the explicit expression for the purchase probabilities under the MNL choice model. Again, we index the universe of n products by the set $\mathcal{N} = \{1, \ldots, n\}$. In addition to these n products, we assume there is an ever-present dummy product with index 0, which is included in each assortment that the retailer potentially could offer. This product is often labeled the no-purchase option and it represents the option for the customer to leave the store without making a purchase. Throughout the paper we assume that $V_{0t} = 0$, which is an assumption that can be made without loss of generality. Under the MNL model, if the retailer offers assortment $S_t \subseteq N$ to customer t, then the probability that product $j \in S_t$ is purchased is given by

$$P_{jt}(S_t, X_t) = \frac{e^{\beta' X_{jt}}}{1 + \sum_{i \in S} e^{\beta' X_{it}}},$$

where $X_t = \{X_{jt} : j \in S_t\}$ gives the features associated with each of the offered products. In this setting, the purchase probabilities depend explicitly on *both* the product/customer features and the set of displayed products. When we move to the assortment problem, the coefficients β will be fixed and we will define $v_{jt} = e^{\beta' X_{jt}}$ to denote the preference weight that customer t associates with product j.

Fitting the MNL choice model. We use maximum likelihood estimation (MLE) to derive estimates for the β coefficients. We formulate the likelihood using historical sales data from τ customers. More specifically, we represent the past purchasing history of the τ customers as the set $\text{PurchaseHistory}_{MNL} = \{(S_t, X_t, z_t) : t = 1, \ldots, \tau\}$, where we note again that S_t denotes the set of six displayed products and $X_t = \{X_{jt} : j \in S_t\}$ gives their associated features. The term z_t gives the product that was purchased, where we set $z_t = 0$ if the customer did not purchase any of the offered products. For customers who purchased multiple products, we treat each purchase independently and hence create a separate data point for each unique product that is purchased. To illustrate how we handle events where an arriving customer makes multiple purchases, we consider a simplified, featureless setting where customer t is offered products $S_t = \{1,2,3\}$ and purchases products 1 and 2. In this case, our purchase history will contain the data points $(\{1,2,3\},1)$ and $(\{1,2,3\},2)$.

With this notation in place, we formulate the MLE problem of interest below

$$\max_{\beta} \mathcal{LL}(\beta \,|\, \text{PurchaseHistory}_{MNL}) \tag{1}$$

where

$$\mathcal{LL}(\beta \,|\, \text{PurchaseHistory}_{MNL}) = \sum_{t=1}^{\tau} \beta' X_{z_t,t} - \log(1 + \sum_{j \in S_t} e^{\beta' X_{j,t}}).$$

In problem (1), the objective is the log-likelihood written as a function of the purchasing history of the τ customers. In the above MLE problem, we seek the β coefficients, which maximize this log-likelihood function. It is well known (see McFadden (1974)) that the objective function in (1) is concave in the β coefficient. Hence, when τ is relatively small, off-the-shelf nonlinear optimization solvers, such as MATLAB's $fmnincon$, are sufficient for solving the MLE problem. For example, Vulcano et al. (2012) and Topaloglu and Simsek (2017) employ this approach to estimate the parameters of an MNL choice model in test cases where τ never exceeds 50,000.

In our setting, we continuously resolve (1) on a rolling week-long basis similar to the machine-learning-based approach, and hence we have $\tau \approx 20 - 30$ million. Further, there is an inherent data censorship issue that results due to no-purchase events, further complicating the estimation process. Recall that when a no-purchase event is observed, we have $z_t = 0$. Unfortunately, it is impossible to know if the arriving customer did not make a purchase because she was not satisfied with the set of offered products or because she never intended to make a purchase in the first place. We refer to customers of this latter type as "browsers." The former scenario provides a signal of how the customer valued the set of offered products, while data from the latter case should be discarded. Consequently, appropriately differentiating between these two cases is critical for deriving accurate estimates of the β values. In our setting, approximately 95% of the observations correspond to no-purchase events, and hence the manner in which this censorship issue is dealt with has nontrivial effects on the accuracy of the estimates produced.

This censorship issue is not new when it comes to solving the estimation problem for various choice models. For example, van Ryzin et al. (2010) and van Ryzin and Vulcano (2017) develop EM algorithms to deal with the brick-and-mortar version of this censorship, in which time periods that have no observed sales are either the result of a no-purchase event or simply the fact that no customer arrived at the store. In this case, an accurate distinction between these two cases is essential for getting an accurate estimate of the probability that a customer arrives in each time period. In theory, these EM-based approaches could be applied in our setting; however, a practical implementation of these algorithms is nearly impossible due to the scale of our problem. In particular, these EM-based approaches rely critically on an efficient way to solve the MLE problem when the censored data is revealed. Further, since EM algorithms are iterative approaches, the resulting "uncensored" or full-information MLE must be solved repeatedly, which is not tractable for the scale of problem we consider.

The above discussion summarizes the two intertwined big-data and censorship difficulties that must be overcome in order to solve problem (1) in our setting. In what follows, we provide a heuristic approach for handling these issues, which we show performs quite well in practice. The steps of this approach unfold as follows:

Step 1: Randomly sample 10% of the no-purchase events.

Step 2: Solve problem (1) using the randomly sampled no-purchase events in addition to all data points (S_t, X_t, z_t), such that $z_t \neq 0$.

Step 3: Scale down each of the estimated β values by a constant δ.

In what follows, we motivate and further explain the implementation details regarding the three steps outlined above. In the first step, we downsample the no-purchase events so problem (1) is reasonably tractable. By discarding 90% of the no-purchase events, we implicitly assume that 90% of customers who do not make a purchase are browsers, which likely is an overestimate of this percentage that we adjust for in step 3. In step 2, we formulate and solve our MLE problem. Even after we downsample the no-purchase events, the optimization problem at hand is still not amenable to commercial nonlinear solvers. Consequently, we solve problem (1) using TensorFlow, which uses a highly parallelized implementation of stochastic gradient ascent. Even with this sophisticated machinery, at least an hour is still required to solve problem (1). Finally, in step 3 we adjust the preference weights of each product to account for the fact that our MNL model is likely fit using a likelihood function that has too few no-purchase events and hence we have overestimated the preference weights of each product. Through extensive out-of-sample testing in which we implement this choice-modeling-based approach for different δ values, we find that setting $\delta = 2000$ is the best scaling coefficient.[4]

[4] In our out-of-sample tests, we try $\delta \in \{0, 500, 1000, 1500, 2000\}$. The MNL-choice-model-based approach is substantially better than the ML approach for all δ. The largest performance difference in terms of average revenue per visit between MNL approaches with different values of δ is less than 5%

3.3. Estimation Case Study: Machine Learning vs. MNL

In this section, we present a case study in which we fit both MNL and machine learning models to historical sales data generated in April 2018 from the coupon sub-pages of the ten most popular sellers on Tmall.com. Due to the fact that we only use sales data from ten sellers to fit our models, the scale of the estimation problems we consider is much smaller than the one encountered within the recommender systems we actually implement on Alibaba. Further, since the exact nature of the machine learning methods used by Alibaba must remain confidential, we are not able to replicate their methods or results exactly in this case study. Instead, we fit machine learning models inspired by the current practice at Alibaba in the sense that both estimation schemes rely on gradient boosted decision trees to estimate the click and purchase probabilities. It is important to note that the intent of this case study is not to perfectly replicate the estimation problem faced by Alibaba, but instead to show that it is relatively straightforward to fit machine learning models that outperform the MNL fits in terms of predictions accuracy. In this way, we shed light on the following fundamental issue that sits at the core of our research: *Machine learning methods are powerful tools for prediction and are often more accurate than MNL models; however, when these predictions seed subsequent optimization problems whose solutions guide key operational decisions, it is not guaranteed that higher prediction accuracy will lead to more profitable decisions.*

Top ten seller statistics. Alibaba has provided us with two weeks of historical sales data from the ten sellers on Tmall.com that experienced the largest volume of traffic in April 2018. We note that this two week time period does not overlap with the time horizon of our field experiments. Table 1 provides an extensive summary of the available sales data for each seller. Further, for each arriving customer t and offered product $t \in S_t$, the feature vector X_{jt} gives the values of the 25 features with the highest importance scores according to the machine learning approaches that have been utilized in the past. Among these top 25 features are product-specific features such as price, the number of good reviews, the number of times the product has been clicked, and the image quality of the associated picture displayed to each customer. In addition, we use customer-specific features such as the given customer's spending and total number of products added to the shopping cart both in the last week and in the last month. Beyond these rather straightforward product/customer features, we also have access to joint features that are specific to each customer and product pair. For example, one such joint feature is the collaborative filtering score signifying the extent to which the particular product will appeal towards the particular customer. Once again, due to confidentiality agreements, we cannot disclose the complete list of all 25 features.

Accuracy metrics and models tested. For each seller, we randomly select 75% of its sales data to be used for fitting the models and hold-out the remaining 25% of the data to test the accuracy of these models. After splitting the data in this way, we aggregate all of the training data from each

Table 1 Key seller statistics

Seller	Product Category	# products	# clicks	# purchases	# customers	conversion %
1	Electronics	169	8,338	2,045	41,765	4.88
2	Women's Apparel	118	17,792	2,163	139,853	1.49
3	Men's Apparel	1,047	11,508	1,956	213,678	0.88
4	Perfume	103	32,535	8,478	131,822	6.16
5	Diapers	132	10,296	2,979	90,467	3.01
6	Furniture	49	4,949	1,937	33,579	5.75
7	Cooking Appliances	38	3,376	2,180	37,925	5.75
8	Cooking Appliances	82	4,220	1,448	40,108	3.59
9	Women's Apparel	501	7,267	2,127	63,466	3.23
10	Bed Linens	115	6,975	1,767	39,494	4.43

Notes. This table reports the key statistics, including categories, number of products and conversion rates, for the top ten sellers that we use for this case study.

seller into a single training set. This set-up most closely resembles the current practice at Alibaba, where the machine learning models are fit to sales data aggregated across all sellers. Once the MNL and machine learning models have been trained, we measure the accuracy of each fitted model using two metrics that are computed using the sales data exclusively from each seller's testing set restricted to customers who purchase exactly one item. In computing these accuracy metrics, we ignore customers who make multiple purchases, which has a negligible effect on our results since multiple products were purchased in approximately 0.01% of customer visits. That said, we defer explanations for why no-purchase events are ignored until the two accuracy metrics are formally defined, since this understanding will help elucidate our choice. The series of steps described above – 75/25 train/test split, fitting the models, computing the accuracy metrics on the test data set – make up what we refer to as a single trial. We eventually present the average accuracy metrics for each seller over 10 trials.

It is important to note that one potential metric that could be used to assess fitting accuracy is the log-likelihood on a hold-out sample of sales data. This metric is often referred to as the out-of-sample log-likelihood, and it has been a popular metric for assessing the accuracy of fitted customer choice models in the revenue management literature (see Topaloglu and Simsek (2017), for example). Unfortunately, comparing the out-of-sample log-likelihoods for the MNL and machine-learning-based approaches would not be an apples-to-apples comparison because the machine learning estimation procedures make predictions at the customer-product level, while the MNL choice model makes predictions at the offer-set level. Consequently, we instead use the following two metrics, which assess how the well the fitted models are able to predict the product that the arriving customer ultimately purchased.

The first metric is the classification accuracy, which is a measure of how frequently we predict correctly the item that is purchased. For each model, this metric is the fraction of customers in the hold-out data set for which the fitted model's predicted purchase probability for the product that was purchased is the largest among all displayed options, excluding the no-purchase option. The reason we ignore the latter option in computing this metric is similar to why we ignore sales data points in the test set that correspond to customers who select the no-purchase option. Essentially, since only 1%-6% of customers made a purchase (see conversion rates in Table 1), all fitted choice models overwhelmingly predict that each customer will select the no-purchase option. As a result, unless the no-purchase option is ignored, there will be little differentiation between the classification accuracy of the fitted models.

The second accuracy metric we compute is referred to as the average rank. For this purpose, we first obtain the purchase probabilities of each displayed option (again, excluding the no-purchase option) under each of the fitted models. Then, for each customer t, we sort the displayed options in order of decreasing purchase probabilities and subsequently find the rank of the purchased product in this sorted list. Our convention is that the product with the largest predicted purchase probability is assigned a rank of 1, the product with the second largest is ranked 2, so on and so forth. With these definitions, the average rank metric is the average rank of the purchased product over all customers in the test set who purchase exactly one product.

Given that we have access to hundreds of thousands of historical data points, even in this simplified case study, it is no simple task to produce accurate estimates of the purchase probabilities in an efficient manner, whether it be by fitting MNL or machine learning models. As a result, the descriptions of the two fitted models below give references to Appendices that provide our exact implementation.

1. **The MNL choice model (MNL):** We fit this model by solving problem (1) via Tensorflow implementation, which is presented in Appendix C.1. We find that in this simplified setting with ten sellers, downsampling the no-purchase events has a negligible effect on the accuracy of the fitted models.
2. **The machine learning models (Trees):** We use gradient boosted classification trees to estimate the click probabilities $\mathbb{P}(C_{jt} = 1)$ and the conditional purchase probabilities $\mathbb{P}(Z_{jt} = 1 | C_{jt} = 1)$. More specifically, we use Catboost (Prokhorenkova et al. 2018), a novel gradient boosting toolkit. The full details of our implementation are given in Appendix C.2.

Results The results for each seller with regards to the two accuracy metrics are presented in Table 2. The first two columns identify the seller number and the fitted model. Columns three and five specify the mean classification accuracy and average rank respectively over 20 trials. Columns four and six correspond to the percentage improvement in performance of the machine learning models over the standard MNL fits. For all ten sellers and both accuracy metrics, the machine learning fits yield statistically significant ($p = 0.05$) improvements over the MNL fits.

Table 2 Predictive performance of the fitted models.

Seller #	Fitted Model	Classification Accuracy	Improvement over MNL	Avg. Rank	Improvement over MNL
1	MNL	0.42	-	1.96	-
1	Trees	0.85	102.38%	1.20	63.33%
2	MNL	0.53	-	2.01	-
2	Trees	0.60	13.21%	1.64	22.56
3	MNL	0.61	-	1.83	-
3	Trees	0.73	19.67%	1.52	20.39%
4	MNL	0.76	-	1.49	-
4	Trees	0.83	9.21%	1.30	14.62%
5	MNL	0.60	-	1.92	-
5	Trees	0.67	11.66%	1.65	16.36%
6	MNL	0.81	-	1.30	-
6	Trees	0.86	6.17%	1.20	8.33%
7	MNL	0.94	-	1.08	-
7	Trees	0.96	2.13%	1.06	1.89%
8	MNL	0.83	-	1.27	-
8	Trees	0.91	9.64%	1.16	9.48%
9	MNL	0.59	-	1.80	-
9	Trees	0.80	35.59%	1.32	36.36%
10	MNL	0.84	-	1.29	-
10	Trees	0.86	2.38%	1.22	5.73%

Notes. This table shows the out-of-sample average classification accuracy and average rank of Machine Learning and MNL models over each of the top ten sellers.

The result in Table 2 clearly show that a simple out-of-the-box machine learning method with minimal parameter tuning is able to outperform the MNL model with regards to both accuracy metrics. Of course, these results are not an exact replica of the accuracy we observe in the experimental setting, however they serve as strong empirical support of the notion that machine learning models have the potential to outperform simpler models in terms of prediction accuracy. However, as we go on to show in our field experiments, this improvement in fitting accuracy does not guarantee that more profitable assortments will be displayed to each arriving customer. In the next section, we detail the subsequent assortment problems that result after the models have been estimated and show why this might be the case.

4. The Assortment Problem

In this section, we consider the assortment problem that results when the estimates of the purchase probabilities from the fitted MNL and machine learning models are used to seed the *Alibaba Product Display Problem*. In the case of the machine learning approach, a simple greedy algorithm is all that is needed to choose the optimal six product displays. In contrast, when the purchase probabilities are dictated by a fitted MNL model, the resulting assortment problem is a cardinality-constrained assortment problem under the MNL choice model. As previously mentioned, a handful of past approaches for solving cardinality-constrained assortment problems exist under the MNL choice model. Since the problem must be solved in an online fashion within a threshold time of 50 milliseconds, we elect to employ a modified version of the combinatorial algorithm of Rusmevichientong et al. (2010), whose running time we are able to improve by a factor of $O(\log n)$. The details of this improved implementation are presented in Appendix A.

4.1. The Machine Learning Fits

After fitting the machine learning models, we are able to derive estimates $P_{jt}(X_{jt})$ of the purchase probabilities for any customer t and product j. Upon the arrival of customer t, the system will first find all products with non-zero inventory and form the set $\mathcal{N}$ from this collection of available products. In this setting, it turns out that the *Alibaba Product Display Problem* can be solved with a straightforward greedy algorithm that first sorts the products in descending order of $r_j \cdot P_{jt}(X_{jt})$ and then selects the top six products in this ordering. This algorithm is trivially optimal because the purchase probabilities do not depend on the set of offered products. Consequently, the problem of choosing the optimal six-product display simply becomes a cardinality-constrained knapsack problem, for which it is straightforward to see that the aforementioned simple greedy algorithm is optimal. Since the six-product displays must be generated in an online fashion for each arriving customer, the simplicity of this optimal greedy approach is to be valued. However, as we go on to demonstrate in Section 6, what is gained in efficiency is lost when sub-optimal product displays are chosen due to the fact that the greedy algorithm chooses to display a particular product without considering how this choice will affect the appeal of the other displayed products.

As discussed at the beginning of this section, another drawback of the machine learning approach is that the estimates of the purchase probabilities should essentially be treated as black-boxes, since the fitted models do not provide a closed form relationships between the features and the predicted purchase probabilities. This is in contrast to the fitted MNL models, for which we assume the deterministic component of the random utility V_{jt} is a linear function of each feature. Consequently, under the fitted machine learning models, it is not possible to formulate an an optimal pricing problem, which could perhaps be one key operational lever for the Alibaba or other retailers to increase revenue.

4.2. The MNL Fits

Next, we consider the cardinality-constrained assortment optimization problem that results when the purchase probabilities P_{jt} in the *Alibaba Product Display Problem* are dictated by our fitted MNL choice model. Again, we consider a setting with n products indexed by the set $\mathcal{N} = \{1, \dots, n\}$, where the revenue of product $j \in N$ is given by r_j. For each customer who arrives, we compute the customer-specific preference weights $v_{jt} = \beta^* X_{jt}$, where β^* is the optimal solution to problem (1), after being scaled down by δ. We encode our assortment decision through the binary vector

$y \in \{0,1\}^n$, where we set $y_j = 1$ if product j is offered and $y_j = 0$ otherwise. The expected revenue of displaying assortment y is denoted as

$$R(y) = \frac{\sum_{j \in N} r_j v_{jt} y_j}{1 + \sum_{i \in N} v_{it}}.$$

Finally, we denote the set of feasible assortments as $\mathcal{F} = \{y \in \{0,1\}^n : \sum_{j=1}^n y_j = 6\}$. Note that the cardinality constraint must be satisfied with equality in our setting, since for each arriving customer we must always display six products. The cardinality-constrained assortment problem of interest can be stated as follows:

$$Z_{OPT} = \max_{y \in \mathcal{F}} R(y). \qquad (\textit{MNL-Card})$$

The first optimal polynomial-time algorithm for problem *MNL-Card* is due to Rusmevichientong et al. (2010). They provide a purely combinatorial approach whose run time is $O(n^2 \log n)$. In Appendix A, we give a novel implementation of this algorithm, which improves upon this previous run time by a factor of $O(\log n)$.

In Appendix B, we consider additional operational levers that could be used by Alibaba to increase revenue in this discount coupon setting. In particular, we consider variations of *Alibaba Product Display Problem* where, on top of product assortments, Alibaba can also control the price or the icon size of each displayed product on the coupon sub-page. We detail how these two constrained versions of the assortment problem under the MNL model can either be solved optimally or near-optimally. For the problem that considers the icon size of each displayed product, we develop a novel approximation scheme.

5. Experiment Design and Data

In this section, we discuss the design of our field experiment. We then provide summary statistics of the raw data as well as the randomization check to demonstrate that our experiment is rigorously conducted.

5.1. Experiment Design

We finished implementing and testing our MNL-choice-model-based approach by the end of February 2018. Recall that the machine-learning-based approach is Alibaba's current practice and hence there was no work to be done in terms of implementing this benchmark approach. Our experiment officially started on March 12, 2018. The field experiment lasted for two weeks, but due to security reasons we can only report the results from the first week (i.e., March 12, 2018, to March 18, 2018).[5]

Throughout the experiment, we test the following three approaches.

[5] The number of customers aggregated across two weeks has surpassed the allowed number of customers to use in a research paper by the company. This is why we focus on the first week of the data. However, our results do not change qualitatively if we use the second week of data.

1. **The MNL-choice-model-based approach (MNL approach):** Customers assigned to this approach see six product displays from the MNL-choice-model-based approach. Similar to the case study presented in Section 3.3, this approach uses the top 25 features based on importance scores in its featurization of the MNL utility functions.
2. **The same-feature-ML-based approach (SF-ML approach):** Customers assigned to this approach see six product displays from the machine-learning-based approach, in which the features used within the machine learning estimation algorithms are the same set of 25 top features.
3. **The all-feature-ML-based approach (AF-ML approach):** Customers assigned to this approach see six product displays from the machine-learning-based approach described in Section 4.1, in which hundreds of features are used within the machine learning estimation algorithms. Before our work, this was the current product recommendation system for choosing the six-product displays on the coupon sub-pages.

During the experimental week beginning on March 12, 2018, each customer who arrives at the coupon sub-pages for any participating seller is randomly assigned one of the three approaches based on a unique hash number derived from the given customer's ID and an experiment ID.[6] Each customer is only assigned to one of the three product recommendation approaches described above regardless of how many times she visits the coupon sub-page.

Given this experimental set-up, we primarily focus on the comparison between the MNL-based approach and the machine-learning-based approach that use the top 25 features. However, as noted in Section 1, we also implemented a full-feature version of our MNL-based approach in September 2018 and compare it with the full-feature machine-learning-based approach in a five-day-long field experiment. The results of this field experiment are presented in Appendix D. We remind the reader that since Alibaba did not provide us with visit-level data for the new experiment, we were not able to explore the accuracy of the full-feature MNL model nor can we conduct heterogeneous treatment analysis.

5.2. Data and Randomization Check

Over the week of our experiment, we observe 27 million customer arrivals from 14 million unique customers. From these 14 million unique customers, we randomly select 5 million to be randomly assigned to one of our three approaches. (The remaining 9 million customers were participants in

[6] To prevent our experiments from colliding with existing experiments on the Alibaba platform, we use a randomization procedure with hashing. In particular, during the experimental week, each arrival customer ID is concatenated with a unique number that is representative of our current experiment. The resulting concatenated number is then hashed into a byte stream using the MD5 message-digest algorithm (Rivest 1992). The first six bytes of this byte stream are extracted and then divided by the largest six-digit hex number to get a floating point. We then assign customers randomly based on this unique floating point value.

Table 3 Summary Statistics

	MNL	SF-ML	AF-ML	Min Pairwise P-value
Panel A: Randomization Check				
Seller Monthly GMV	1.7 million	1.7 million	1.7 million	> 0.3
Seller Number of Products	2187	2186	2189	> 0.2
Seller Registration Year	2013	2013	2013	> 0.4
Customer Registration Year	2012	2012	2012	> 0.3
Customer Gender (Male =1)	0.26	0.26	0.26	> 0.5
Customer Age	30.2	30.2	30.3	> 0.3
Panel B: Summary Statistics				
Number of Page Views	3,469,129	3,484,555	3,467,965	
Number of Products Clicked	421,896	368,987	423,046	
Number of Products Purchased	86,585	70,699	90,033	
GMV (RMB)	18 million	14 million	17.8 million	

Notes. Panel A reports the average monthly GMV, average number of products available to the seller, average seller registration year, and average customer registration year, customer gender breakdown and average age for all sellers and customers assigned to each approach (i.e., MNL, SF-ML and AF-ML approach). T-tests between the differences in averages of the three approaches have $p-value$ greater than 0.05 for all pair-wise comparisons. Panel B reports the total number of page views, number of products clicked/purchased and total GMV in each approach.

other parallel experiments.) In particular, 1,879,903 customers are assigned to the MNL approach, 1,879,598 customers are assigned to the SF-ML approach, and 1,876,940 customers are assigned to the AF-ML approach. These 5 million unique customers generate 10 million arrivals to coupon sub-pages during the week of our experiment. (Given that our experiment relied on the unique experiment ID in hashing, there were no other major experiments during this time that collided with our experiment.)

Next, we present customer and seller information from the three experiment groups to confirm that the customers and sellers assigned to each of the three approaches are comparable in terms of demographics, spending habits, and revenue. Panel A of Table 3 shows the averages of the total GMV in the month prior to the starting date of the experiment; the number of active products on March 12, 2018; registration year; customer age; customer gender breakdown; and customer registration year for each of the three approaches. It is clear that customers and sellers assigned to each of the three approaches have statistically indistinguishable metrics: the minimum p-value over all t-tests is greater than 0.2. The results of our randomization checks suggest that any difference between customers under these three approaches after the experiment was implemented should be attributed to differences in the estimation and assortment algorithms implemented within each approach.

Panel B of Table 3 shows the aggregate impressions made by the arriving customers. More specifically, this table shows that the customers in our experiment generated 3,469,129, 3,484,555, and 3,467,965 page views under the MNL, SF-ML, and AF-ML approaches respectively. This means that on average, each customer viewed approximately 1.85 coupon sub-pages during the week of our experiment. Customers assigned to the MNL approach clicked on $421,896$ displayed products, while customers assigned to the SF-ML and AF-ML approaches clicked on $368,987$ and $423,046$ displayed products respectively. Further, customers assigned to the MNL, SF-ML, and AF-ML approaches respectively purchased $86,585$, $70,699$, and $90,033$ products, leading to RMB 18, 14, and 17.8 million (equivalent to USD 2.63, 2.05 and 2.60 million) respectively. These preliminary results suggest that on average, customers assigned to the MNL approach generated more revenue compared to those assigned to the SF-ML and AF-ML approaches.

6. Main Results

In this section, we present the results of our field experiment. We begin by detailing the financial performance of the three approaches. The metric Alibaba uses internally to assess the profitability of product recommendation systems is GMV (used synonymously with revenue throughout) per customer visit, and hence we also adopt this metric as our means of judging the efficacy of the three approaches. After presenting these results, we dig deeper into the data in an attempt to better understand why some approaches perform better than others. First, we present the accuracy of the purchase probability estimates under each approach. One would expect the approaches with more accurate estimation schemes to perform better, but this might not always be the case. We then present the average price of the products purchased under each approach; we find that the MNL approach recommends six-product displays that lead to more sales of profitable products. Lastly, we document how the performance differences among approaches may change with respect to differences in seller characteristics. In presenting these results, the unit of analysis is the customer t who visited the coupon sub-page of seller k.

6.1. Financial Performance

We begin by presenting the GMV per customer visit generated by each of the three approaches. We define $\text{RevenuePerVisit}_{kt}$ to be the revenue generated from customer t's visit to the coupon sub-page of seller k. Panel A of Table 4 shows the revenue per visit of the MNL, SF-ML, and AF-ML approaches during our experimental period. The first row of Panel A shows that the MNL, SF-ML, and AF-ML approaches generate RMB 5.17, 4.04, and 5.16 per customer visit respectively (equivalent to USD 0.768, 0.600, and 0.767). The revenue per visit under the MNL approach is RMB 1.13, or 28% larger than the revenue per visit under the SF-ML approach. Both the t-test and the nonparametric Wilcoxon test show that this difference is highly significant (all p-values < 0.0001).

While the MNL approach significantly outperforms the SF-ML approach, its financial performance surprisingly is also on par with that of the AF-ML approach, which uses hundreds of features

Table 4 Model Financial Performance

Panel A: Summary Statistics of Financial Performance				
	MNL	SF-ML	MNL	AF-ML
RevenuePerVisit (RMB)	5.17	4.04	5.17	5.16
Difference (All p-values)	1.13 (< 0.0001)		0.01 (0.8346)	
Relative Improvement	28.0%		0.2%	
Observations	3,469,129	3,484,555	3,469,129	3,467,965

Panel B: OLS Regression Results on Model Financial Performance		
	Dependent variable:	
	Revenue	Revenue
	(1)	(2)
SF-ML	-1.126****	-0.987****
	(0.094)	(0.073)
AF-ML	-0.015	0.032
	(0.110)	(0.077)
Customer Controls	No	Yes
Seller Fixed Effect	No	Yes
Date Fixed Effect	No	Yes
Observations	10,421,649	10,421,649

Notes. $*p < 0.10; **p < 0.05; ***p < 0.01; ****p < 0.001$. Standard errors are robust and clustered at the customer level. Panel A reports the average financial performance, in terms of revenue per customer visit, across different algorithms during our experimental period (March 12, 2018 - March 18, 2018). Panel B reports the results from OLS regression that estimate the difference between different models' revenue per customer visit. Column (1) of Panel B does not control for any additional control variables, while Column (2) of Panel B controls for customer characteristics, seller fixed effects and date fixed effects.

within its estimation scheme compared to the 25 features used within the MNL approach. Both the t-test and the nonparametric Wilcoxon test show that the financial performance difference with respect to revenue generated per visit between these two approaches is not statistically significant (all p-values > 0.8346). This result shows the potential of the MNL approach: using only a small fraction of the features used by the AF-ML approach, our MNL approach can generate similar revenue per customer visit. This result leads us to believe that we would observe a sizable improvement if we extended the MNL approach to include all features. Alibaba has indeed indicated to us that they would like to prioritize implementing the MNL-based approach using all of their available features.

Next, we test the differences in financial performance with respect to revenue generated per visit between the three approaches, controlling for specific customer and seller characteristics that may affect customer spending behavior. Since this is a field experiment with proper randomization, control variables are added only to make the estimators more efficient. Specifically, we use the following OLS regression specification:

$$\text{RevenuePerVisit}_{kt} = \alpha_0^1 + \alpha_1^1 \text{Approach}_t + X_t + X_k + D_t + \epsilon_{kt}. \quad (2)$$

In the expression above, Approach_t is a categorical variable indicating the approach to which customer t has been assigned. The terms X_t and X_k represent customer- and seller-specific features, including customer age, customer gender, customer registration year, the seller's GMV from the previous month, the seller's registration year, the category of products sold by the seller, and the number of products the particular seller offers. The term D_t gives a date-specific fixed effect. We report the robust standard errors clustered at the customer level in this analysis as well as all subsequent analyses presented in this paper. All of our findings continue to hold if we cluster standard errors at both the customer and seller levels.

Panel B of Table 4 gives the results from specification (2). In this specification, we use data from the MNL approach as the baseline, so the coefficients of SF-ML and AF-ML approach indicators represent the financial performance difference between the MNL approach and each of the other two approaches. Column (1) of Panel B does not control for any additional variables, and we successfully recover the mean difference from Panel A: a customer visit under the MNL approach generates 1.126 and 0.015 more RMB per visit compared to the SF-ML and AF-ML approaches. The difference between the financial performance of the MNL approach and the SF-ML approach is statistically significant, while the financial performance of the MNL approach is statistically indifferent from that of the AF-ML approach. Column (2) of Panel B controls for the customer characteristics, seller fixed effects, and date fixed effects, and qualitatively we observe the same set of results.

The results described above indicate that the MNL approach performs quite well in relation to both of the machine-learning-based approaches. In what follows, we show that this superior performance cannot be explained by superior prediction accuracy, since similar to the results of our case study presented in Section 3.3, we find again that Alibaba's machine learning models are far more accurate than the fitted MNL models. As such, we subsequently provide an alternative explanation for why the MNL-based-approach performs so well, and also explore where there is potential for improvement with regard to the MNL approach.

6.2. Purchase Probability Accuracy

In this section, we comparing the fitting accuracies of each fitted model using the experimental sales data that was generated from March 12, 2018, to March 18, 2018, as the hold-out data set from which we compute the two accuracy metrics described in Section 3.3. Due to the complexities of implementing various approaches, each customer can only be assigned to a single approach; therefore, for each customer visit we only have access to one set of purchase probability estimates. Moreover, given the historical data of each customer visit, we cannot retrospectively compute the purchase probabilities from other approaches that were not used to serve this visit because the

Table 5 Model Prediction Performance

Panel A: Summary Statistics of Prediction Performance on Purchases				
	MNL	SF-ML	MNL	AF-ML
ClassificationAccuracy	36.31%	74.55%	36.31%	77.50%
Difference (All p-values)	38.24% (< 0.0001)		41.19% (< 0.0001)	
AverageeRank	2.51	1.51	2.51	1.43
Difference (All p-values)	1.00 (< 0.0001)		1.08 (< 0.0001)	
Observations	82,957	68,395	82,957	86,238

Panel B: OLS Regression Results on Model Prediction Performance		
	Dependent variable:	
	ClassificationAccuracy	AverageRank
	(1)	(2)
SF-ML	0.408^{****}	-1.075^{****}
	(0.003)	(0.007)
AF-ML	0.437^{****}	-1.152^{****}
	(0.003)	(0.006)
Buyer Controls	Yes	Yes
Seller Fixed Effect	Yes	Yes
Date Fixed Effect	Yes	Yes
Observations	237,417	237,417

Note: $*p < 0.10; **p < 0.05; ***p < 0.01; ****p < 0.001$. Standard errors are robust and clustered at the customer level. Panel B reports the average prediction power of customers' purchasing behaviors during our experiment. In Panel C, Columns (1) and (2) report the reports from OLS regression on models' prediction power of customers' purchasing behaviors.

dynamic features of this visit cannot be recorded by the system. As a result, for each approach we only compute the two accuracy metrics using data from customers who were assigned to that particular approach, which might result in accuracy scores that are biased by the underlying approach used to choose the six-product displays. For example, instead of solving problem *MNL-Card*, imagine that the MNL approach always recommended the product with the largest preference weight along with the five products with lowest preference weights. This approach is not likely to be profitable, but it will lead to a high classification accuracy for the MNL approach.

The discussion above explains why there is a different number of observations for each approach in Table 5. Further, this discussion is included for full transparency. While our results clearly are not biased to the extent of the example above, they should nonetheless be taken with a grain of salt; the results have qualitative significance, but the exact accuracy scores might not perfectly reflect the accuracy of the underlying estimation scheme. Again, the complexities and intricacies that come with implementing this sort of full-scale recommendation system significantly complicate any sort of ex-post analysis.

Panel A of Table 5 gives the accuracy of the fitted model for each of the three approaches based on the two metrics we consider. The top two rows of Panel A show that the classification accuracies are 36.31%, 74.55%, and 77.50% for the MNL, SF-ML, and AF-ML approaches respectively. These differences in classification accuracies are highly significantly (all p-values < 0.0001). The bottom two rows of Panel A show that the average rank of the purchased products is 2.51, 1.51, and 1.43 for the MNL, SF-ML, and AF-ML approaches respectively and that the pair-wise differences between these average ranks are statistically significantly (all p-values < 0.0001).

We next conduct regression analyses to determine whether the differences in prediction accuracy can be recovered when we control for characteristics that may affect customer purchasing behavior. We introduce two terms to conduct this analysis: TopPurchased_{kt} and $\text{AveragePurchaseRank}_{kt}$. TopPurchased_{kt} is a binary indicator that is 1 if customer t who visited seller k purchased the product with the highest predicted purchase probability and 0 otherwise. $\text{AveragePurchaseRank}_{kt}$ is the average rank of purchased products for customer t's visit to seller k. We use the following OLS regression specification:

$$\text{TopPurchased}_{kt} = \alpha_0^2 + \alpha_1^2 \text{Approach}_t + X_t + X_k + D_t + \epsilon_{kt} \tag{3}$$

$$\text{AveragePurchaseRank}_{kt} = \alpha_0^3 + \alpha_1^3 \text{Approach}_t + X_t + X_k + D_t + \epsilon_{kt} \tag{4}$$

where the set of controls is the same as in specification (2). Our results all hold true if we cluster standard errors at both the customer and seller levels or employ a logistic regression on TopPurchased_{it} (a binary dependent variable).

Columns (1) and (2) in Panel B of Table 5 present results from specifications (4) and (5). In these specifications, we use the accuracy performance under the MNL approach as a baseline, so the coefficients of the SF-ML and AF-ML indicators represent the difference between the MNL approach and each of the machine learning approaches. The coefficients of column (1) are all positively significant, showing that both machine learning approaches have higher prediction accuracy compared to the MNL approach. Notice that the magnitude of the difference (for example, 29.8% between the MNL approach and the SF-ML approach) is similar to that in Panel A (i.e., 28.43%). This shows that controlling for additional fixed effects does not change our results much, which provides further evidence that our experiments are properly randomized. Column (2) echoes this result by showing that the average rank of purchased products under both machine learning approaches is lower than the average rank of the purchased products under the MNL approach.

As foreshadowed by the case study presented in Section 3.3, we demonstrate that while the MNL approach performs much better than the SF-ML approach and on par with the AF-ML approach in terms of revenue per visit, it actually has significantly worse prediction accuracy than both machine learning approaches with respect to both accuracy metrics. Consequently, we seek an explanation for the superior financial performance of the MNL approach. In what follows, we provide one such explanation: the MNL approach produces six-product displays that ultimately lead to higher revenue products being purchased.

Table 6 Mechanism Behind MNL-based Model's Superior Financial Performance

Panel A: Summary Statistics of Mechanisms				
	MNL	SF-ML	MNL	AF-ML
RevenuePerVisit (RMB)	5.17	4.04	5.17	5.16
Difference (All p-values)	1.13 (< 0.0001)		0.01 (0.8346)	
PurchaseIncidence	2.39%	1.96%	2.39%	2.49%
Difference (All p-values)	0.43 (< 0.0001)		-0.1 (< 0.0001)	
Observations	3,469,129	3,484,555	3,469,129	3,467,965
PricePerPurchase	216.2	206.1	216.2	207.3
Difference (All p-values)	10.1 (< 0.0001)		9.9 (< 0.0001)	
Observations	82,957	68,395	82,957	86,238

Panel B: OLS Regression Results on Model Financial Performance				
	Dependent variable:			
	Revenue		PurchaseIncidence	PricePerPurchase
	(1)		(2)	(3)
SF-ML	-0.987^{***}		-0.004^{***}	-6.349^{***}
	(0.073)		(0.0001)	(1.820)
AF-ML	0.032		0.001^{***}	-5.895^{***}
	(0.077)		(0.0001)	(2.069)
Buyer Controls	Yes	Yes	Yes	Yes
Seller Fixed Effect	Yes	Yes	Yes	Yes
Date Fixed Effect	Yes	Yes	Yes	Yes
Observations	10,410,587		10,410,587	237,417

Note: $*p < 0.10; **p < 0.05; ***p < 0.01; ****p < 0.001$. Standard errors are robust and clustered at the customer level. Panel A reports the average revenue per visit, average purchasing probability and average price conditional on purchasing across different algorithms duing our experiment, March 12, 2018 to March 18, 2018. Panel B report the corresponding results from OLS regressions controlling for customer characteristics, seller fixed effects and date fixed effects.

6.3. Average Purchase Price

In this section, we provide one potential explanation for the superior financial performance of the MNL approach. Specifically, we show that on average the MNL-based approach chooses six-product displays that lead to purchases of higher revenue products. To formalize this analysis, we first define $\text{PurchaseIncidence}_{kt}$ as a binary indicator equal to 1 if customer t's visit to seller k results in a purchase and 0 otherwise. We also define $\text{PricePerPurchase}_{kt}$ as the average price of the purchased products during customer t's visit to seller k.

Panel A of Table 6 shows the $\text{RevenuePerVisit}_{kt}$, $\text{PurchaseIncidence}_{kt}$, and $\text{PricePerPurchase}_{kt}$ for all three approaches during our experimental period. The left side of Panel A shows that the MNL approach generates a higher revenue per visit and has a higher purchasing incidence than the SF-ML approach. In particular, under the MNL approach the average purchasing price is 4.9% higher than the average purchasing price under the SF-ML approach. Further, under the MNL approach customers on average make a purchase 22% more frequently than under the SF-ML approach. Hence, while the SF-ML approach produces more accurate estimates of the purchase probabilities, it is not able to offer assortments that are as desirable nor as profitable as those offered by the MNL approach.

The comparison between the MNL approach and AF-ML approach is shown on the right side of Panel A in Table 6. We see that the MNL approach leads to a significantly higher average purchasing price (i.e., RMB 216.2 versus RMB 207.3, p-value < 0.00001) and significantly lower purchasing incidence (i.e., 2.39% versus 2.49%, p-value < 0.00001), which ultimately leads to similar revenue performance as the two metrics balance each other. It is interesting that there is only a small, albeit statistically significant (p-value < 0.01) improvement in the accuracy of the estimated purchase probabilities as we move from AF-ML to SF-ML, but there is a large improvement in the revenue per visit (also statistically significant). This either demonstrates that the efficacy of the machine-learning-based approaches is highly sensitive to the accuracy of the estimated purchase probabilities or it shows that additional accuracy metrics are needed to better tease out the differences in the estimated purchase probabilities under the two approaches. Panel B of Table 6 reports the regression results, controlling for customer characteristics, seller fixed effects, and date fixed effects and using specifications similar to specification 2. The regression results generate the same insights as those in Panel A of Table 6.

6.4. Heterogeneous Treatment Effect and Weakness of the MNL-Based Approach

In this section, we present several exploratory analyses about the heterogeneous treatment effects of using the MNL approach versus the machine-learning-based approaches. There are two salient limitations of using the MNL choice model to capture customer purchasing patterns in this setting. First, the MNL choice model assumes that each customer only buys a single product, while in practice customers often make multiple purchases. Second, the MNL choice model in its standard form cannot incorporate customer click behavior within how it models customer preferences. Based on these theoretical limitations, we identify two seller characteristics that may influence the performance of the MNL approach. We first define $\text{MultiPurchaseCount}_k$ to be the number of visits to seller k in which the customer makes multiple purchases. Second, we define $\text{Click-to-Purchase}_k$ as the ratio of the number of clicked products to the number of purchased products across all visits to seller k.

We rely on the following OLS regression specifications to test the interaction between the algorithm indicator and the aforementioned list of moderating factors:

$$\begin{aligned}\text{RevenuePerVisit}_{kt} =& \alpha_0^4 + \alpha_1^4 \text{Approach}_t + \alpha_2^4 \text{Moderating Factor}_k + \\ & \alpha_3^4 \text{Approach}_t \times \text{ModeratingFactor}_k + X_t + X_k + D_t + \epsilon_{kt}\end{aligned} \tag{5}$$

Table 7 Heterogeneous Treatment Effect

	Dependent variable:	
	Revenue	
	(1)	(2)
SF-ML	-0.822^{****}	-0.780^{****}
	(0.068)	(0.065)
SF-ML ×MultiPurchaseCount	0.030^{***}	
	(0.012)	
SF-ML ×Click-to-Purchase		0.067^{*}
		(0.035)
Customer Controls	Yes	Yes
Seller Fixed Effects	No	No
Date Fixed Effects	Yes	Yes
Observations	5,326,664	5,326,664

Note: $*p < 0.10; **p < 0.05; ***p < 0.01; ****p < 0.001$. Standard errors are robust and clustered at the customer level. This table reports the results based on Equation 5.

where $\text{ModeratingFactor}_i \in \{\text{Click-to-Purchase}_k, \text{MultiPurchaseCount}_k\}$. We only focus on the observations corresponding to customers who were assigned to the MNL and SF-ML approaches and from sellers who had at least 100 visits, who make up more than 76% of the sellers.[7]

Table 7 reports the results of our heterogeneous treatment analyses. Column (1) of Table 7 shows that the coefficient of the interaction of the SF-ML indicator and MultiPurchaseCount is positive, demonstrating that the difference in financial performance of our MNL approach and the SF-ML approach shrinks when there are more multiple-purchase incidences. In other words, our MNL approach performs worse for sellers whose customers are more likely to purchase multiple items from an offer set. Column (2) also demonstrates a positive interaction term between the SF-ML indicator and Click-to-Purchase. Similarly, this demonstrates that the MNL-based approach performs worse when the ratio of clicks to purchases is high. Columns (1) and (2) collectively show that the theoretical limitations we identified with the MNL approach indeed affect its performance: a one unit increase in $\text{MultiPurchaseCount}_k$ ($\text{Click-to-purchase}_k$) would translate to 0.03 (0.067) decrease in the financial performance differences between the MNL and SF-ML approaches. This is a sign that we need future work to build choice models that are able to model click behavior as well as customers buying multiple items from a set of offered products.

7. Discussion and Conclusion

In this paper, to the best of our knowledge, we document the first full-scale implementation of a customer-choice-model-based product recommendation system. We find that our MNL-based approach generates 28% higher revenue per customer visit compared to the machine-learning-based

[7] Our results are qualitatively robust if we use other numbers of visit cutoffs, such as 500 or 1000.

approach that uses the same set of features. Moreover, we find that our MNL-based approach performs slightly better than the current full-feature machine-learning-based approach, which Alibaba has been improving for more than two years. We then show that our MNL approach performs well because it recommends more profitable items by incorporating substitution behavior within the operational problem that guides product display decisions. However, while the machine learning approach can leverage big data to produce accurate estimates of the purchase probabilities, it sometimes fails to identify profitable sets of products to display because it does not factor in the substitution behavior between products offered together. We are hopeful that our work inspires other companies to consider a choice-model-based approach within their product recommendation system and that it also encourages other researchers in operations management to seek out avenues to implement their algorithms in practice.

In order to further improve choice-model-based recommendation systems, we again highlight several main difficulties in implementing our MNL-based approach, and in doing so we also shed light on several potential research directions. The first difficulty we faced was properly dealing with the inherent censorship issue present in the sales data in the process of estimating our MNL model. Previous techniques used to uncensor the data are rendered ineffective for the scale of problems common in industry, which calls for future research to deal with censorship in a big data setting. Second, we show that our MNL-based approach does not perform well when customers purchase multiple items from the offer set. Unfortunately, to the best of our knowledge there is no choice model that is able to capture multiple purchase events from a single customer visit. Developing such a model represents a natural next step to expand the breadth of retailing scenarios that can be captured using choice models. Lastly, in Section 6.4, we demonstrate that the click data potentially provides useful signals of customer preference. Since our MNL-based approach ignores click behavior in both the estimation and assortment phases, one interesting direction may consider incorporating click behavior within the MNL choice model framework.

References

Adelman, Daniel. 2007. Dynamic bid prices in revenue management. *Operations Research* **55**(4) 647–661.

Breese, John S, David Heckerman, Carl Kadie. 1998. Empirical analysis of predictive algorithms for collaborative filtering. *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 43–52.

Cachon, Gérard P, Santiago Gallino, Marcelo Olivares. 2018. Does adding inventory increase sales? evidence of a scarcity effect in us automobile dealerships. *Management Science* .

Caro, Felipe, Jérémie Gallien. 2007. Dynamic assortment with demand learning for seasonal consumer goods. *Management Science* **53**(2) 276–292.

Caro, Felipe, Jérémie Gallien. 2010. Inventory management of a fast-fashion retail network. *Operations Research* **58**(2) 257–273.

Covington, Paul, Jay Adams, Emre Sargin. 2016. Deep neural networks for youtube recommendations. *Proceedings of the 10th ACM conference on recommender systems*. ACM, 191–198.

Davis, James, Guillermo Gallego, Huseyin Topaloglu. 2013. Assortment planning under the multinomial logit model with totally unimodular constraint structures. *Department of IEOR, Columbia University. Available at http://www. columbia. edu/gmg2/logit const. pdf* 335–357.

Davis, James M, Guillermo Gallego, Huseyin Topaloglu. 2014. Assortment optimization under variants of the nested logit model. *Operations Research* **62**(2) 250–273.

Désir, Antoine, Vineet Goyal, Jiawei Zhang. 2014. Near-optimal algorithms for capacity constrained assortment optimization. Working paper, available as SSRN report #2543309.

Feldman, J., H. Topaloglu. 2017. Assortment optimization under the multinomial logit model with nested consideration sets. *Operations Research* .

Feldman, Jacob, Alice Paul. 2017. Relating the approximability of the fixed cost and space constrained assortment problems. *Production and Operations Management* .

Feldman, Jacob, Huseyin Topaloglu. 2015. Capacity constraints across nests in assortment optimization under the nested logit model. *Operations Research* **812-822** 63.

Ferreira, Kris Johnson, Bin Hong Alex Lee, David Simchi-Levi. 2015. Analytics for an online retailer: Demand forecasting and price optimization. *Manufacturing & Service Operations Management* **18**(1) 69–88.

Friedman, Jerome H. 2002. Stochastic gradient boosting. *Computational Statistics & Data Analysis* **38**(4) 367–378.

Gallego, G., G. Iyengar, R. Phillips, A. Dubey. 2004. Managing flexible products on a network. Tech. Rep. TR-2004-01, Computational Optimization Research Center, Columbia University.

Gallego, G., H. Topaloglu. 2014. Constrained assortment optimization for the nested logit model. *Management Science* **60** 2583–2601.

Gallego, Guillermo, Anran Li, Van-Anh Truong, Xinshang Wang. 2016. Approximation algorithms for product framing and pricing. *Submitted to Operations Research* .

Gallego, Guillermo, Richard Ratliff, Sergey Shebalov. 2011. A general attraction model and an efficient formulation for the network revenue management problem. Tech. rep., Technical report, Columbia University, New York, NY.

Golrezaei, Negin, Hamid Nazerzadeh, Paat Rusmevichientong. 2014. Real-time optimization of personalized assortments. *Management Science* **60**(6) 1532–1551.

Jahrer, Michael, Andreas Töscher, Robert Legenstein. 2010. Combining predictions for accurate recommender systems. *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 693–702.

LeCun, Yann, Yoshua Bengio, Geoffrey Hinton. 2015. Deep learning. *nature* **521**(7553) 436.

Li, G., P. Rusmevichientong, H. Topaloglu. 2015. The d-level nested logit model: Assortment and price optimization problems. *Operations Research* **63** 325–342.

Linden, Greg, Brent Smith, Jeremy York. 2003. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing* (1) 76–80.

Luce, R. D. 1959. Individual choice behavior: A theoretical analysis. *Frontiers in Econometrics* **2** 105–142.

McFadden, Daniel. 1974. Conditional logit analysis of qualitative choice behavior. *Frontiers in Econometrics* **2** 105–142.

McFadden, Daniel, Kenneth Train. 2000. Mixed mnl models for discrete response. *Journal of Applied Econometrics* 447–470.

Papanastasiou, Yiangos, Nicos Savva. 2016. Dynamic pricing in the presence of social learning and strategic consumers. *Management Science* **63**(4) 919–939.

Prokhorenkova, Liudmila, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, Andrey Gulin. 2018. Catboost: unbiased boosting with categorical features. *Advances in Neural Information Processing Systems*. 6639–6649.

Ravikumar, Pradeep, Martin J Wainwright, John D Lafferty, et al. 2010. High-dimensional ising model selection using 1-regularized logistic regression. *The Annals of Statistics* **38**(3) 1287–1319.

Ricci, Francesco, Lior Rokach, Bracha Shapira. 2011. Introduction to recommender systems handbook. *Recommender systems handbook*. Springer, 1–35.

Rivest, Ronald. 1992. The md5 message-digest algorithm. Tech. rep.

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

Rusmevichientong, Paat, Zuo-Jun Max Shen, David B. Shmoys. 2010. Dynamic assortment optimization with a multinomial logit choice model and capacity constraint. *Operations Research* **58**(6) 1666–1680.

Subramanian, Lakshminarayanan, Ashwin Venkataraman, Srikanth Jagabathula. 2018. A conditional gradient approach for nonparametric estimation of mixtures of choice models. unpublished manuscript.

Talluri, Kalyan, Garrett Van Ryzin. 1998. An analysis of bid-price controls for network revenue management. *Management science* **44**(11-part-1) 1577–1593.

Talluri, Kalyan, Garrett Van Ryzin. 1999. A randomized linear programming method for computing network bid prices. *Transportation science* **33**(2) 207–216.

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

Talluri, Kalyan T. 2010. A randomized concave programming method for choice network revenue management .

Topaloglu, Huseyin. 2009. Using lagrangian relaxation to compute capacity-dependent bid prices in network revenue management. *Operations Research* **57**(3) 637–649.

Topaloglu, Huseyin, Serdar Simsek. 2017. Technical note—an expectation-maximization algorithm to estimate the parameters of the markov chain choice model. *Operations Research* Forthcoming.

Train, Kenneth. 2009. *Discrete Choice Methods with Simulation*. Cambridge University Press.

van Ryzin, G., G. Vulcano, W. Chaar. 2010. Choice-based revenue management: An empirical study of estimation and optimization. *Manufacturing and Service Operations Management* **12** 371392.

van Ryzin, Garrett, Gustavo Vulcano. 2017. Technical note—an expectation-maximization method to estimate a rank-based choice model of demand. *Operations Research* **65**(2) 396–407.

Vulcano, Gustavo, Tarik Abdullah. 2018. Demand estimation under the multinomial logit model from sales transaction data. unpublished manuscript.

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

Zhang, Dan, Daniel Adelman. 2009. An approximate dynamic programming approach to network revenue management with customer choice. *Transportation Science* **43**(3) 381–394.

Zhang, Dan, William L Cooper. 2005. Revenue management for parallel flights with customer-choice behavior. *Operations Research* **53**(3) 415–431.

Appendices

A. An Improved Algorithm for the Cardinality Constrained MNL Assortment Problem

The cardinality-constrained assortment problem of interest can be stated as follows:

$$Z_{OPT} = \max_{y \in \mathcal{F}} R(y). \qquad (MNL\text{-}Card)$$

We let y^* be the optimal solution to problem *MNL-Card*. In what follows, we give a novel implementation of this algorithm, which improves upon this previous run time by a factor of $O(\log n)$.

First, following the direction of Rusmevichientong et al. (2010), we consider the function

$$f(z) = \max_{y \in \mathcal{F}} \sum_{j \in N} v_j (r_j - z) y_j. \qquad (6)$$

Additionally, we let

$$\hat{y}(z) = \arg\max_{y \in \mathcal{F}} \sum_{j \in N} v_j (r_j - z) y_j.$$

We note that for a fixed z, it is fairly straightforward to recover $\hat{y}(z)$. To see this, let $c_j(z) = v_j(r_j - z)$ be the "contribution" of product $j \in N$ to the objective value of (6). The assortment $\hat{y}(z)$ will trivially consist of the six products with the largest values of $c_j(z)$. The following theorem, which has appeared in one form or another in numerous assortment optimization papers (Rusmevichientong et al. 2010, Davis et al. 2014, 2013), elucidates the importance of (6).

Theorem 1. *Let $\hat{z} \geq 0$ satisfy $f(\hat{z}) = \hat{z}$, then we have that $R(\hat{y}(\hat{z})) = R(y^*)$.*

T o help unclutter notation, we drop the dependence of $\hat{y}(\hat{z})$ on $\hat{z}$ and simply write $\hat{y}$ for the remainder of this proof. First, we note that such a fixed point must exist, since $f(0) \geq 0$ and $f(z)$ is decreasing in z. We begin by showing that $R(\hat{y}) = \hat{z}$. To see this, note that $f(\hat{z}) = \hat{z}$ implies that

$$\begin{aligned}
&\sum_{j \in N} v_{jt}(r_j - \hat{z})\hat{y}_j = \hat{z} \\
&\implies \sum_{j \in N} v_{jt} r_j \hat{y}_j = \hat{z}(1 + \sum_{j \in N} v_{jt}) \\
&\implies R(\hat{y}) = \hat{z},
\end{aligned}$$

where the final implication results by dividing both sides of the equality by $1 + \sum_{j \in N} v_{jt}$. Next we show that $R(y^*) \leq \hat{z}$. Since $f(\hat{z}) = \hat{z}$ and y^* is feasible to problem (6), we have that

$$\begin{aligned}
&\sum_{j \in N} v_{jt}(r_j - \hat{z})y^*_j \leq \hat{z} \\
&\implies \sum_{j \in N} v_{jt} r_j y^*_j \leq \hat{z}(1 + \sum_{j \in N} v_{jt}) \\
&\implies R(y^*) \leq \hat{z}.
\end{aligned}$$

Finally, combining the two results gives that $R(y^*) \leq \hat{z} = R(\hat{y})$, and hence since y^* is optimal to *MNL-Card*, we must have $R(\hat{y}) = R(y^*)$.

In short, Theorem 1 states that if we can find a $\hat{z} \geq 0$ that is a fixed point of (6), then we can recover the optimal assortment to problem *MNL-Card* through $\hat{y}(\hat{z})$. First, such a fixed point is guaranteed to exist since $f(0) \geq 0$ and $f(z)$ is trivially decreasing in z. Hence, all that remains is to describe an efficient process for finding $\hat{z}$. The most naive (and impractical) way to accomplish this task would be to check all possible values of z. Equivalently, we could compute all of the unique assortments $\hat{y}(z)$ that result from this search over all possible values of z and then select the one with the largest expected revenue. At first glance, this approach seems as equally impractical as the search over all possible values of z. However, Rusmevichientong et al. (2010) cleverly note that since the assortments $\hat{y}(z)$ only depend on the relative ordering of the contributions of each product $c_j(z)$, the number of unique assortments that can possibly arise from an exhaustive search over all possible values of z is $O(n^2)$. To see this, note that the relative ordering of the contributions $c_j(z)$

only changes at values of z where $c_i(z) = c_j(z)$ for products $i, j \in N$. Further, since the contribution of each product is a linear function of z, there can be at most $O(n^2)$ intersection points since each of the n lines can intersect one of the other $n-1$ lines at most once.

Algorithm 1 Median Bisection

t=0
$z_- \leftarrow 0$
$z_+ \leftarrow \max_{i \in N} r_i$
$\bar{Z}_t \leftarrow \mathcal{Z}$
while $\bar{Z}_t > 2$ **do**
    $z_m \leftarrow Med(\bar{Z}_t)$
    $f \leftarrow f(z_m)$
    if $f < z$ **then**
        $z_- \leftarrow z_m$
        $\bar{Z}_{t+1} \leftarrow \{z \in \bar{Z}_t : z \geq z_m\}$.
    else
        $z_+ \leftarrow z_m$
        $\bar{Z}_{t+1} \leftarrow \{z \in \bar{Z}_t : z < z_m\}$
    end if
    $t \leftarrow t+1$
end while
return $\hat{y}(z_- + \frac{z_+ - z_-}{2})$

More formally, for products $i, j \in N$, we let $z(i,j)$ be the value of z satisfying $c_i(z) = c_j(z)$. In other words, $z(i,j) = \frac{v_i r_i - v_j r_j}{v_i - v_j}$. We denote the set of all such intersection points as $\mathcal{Z} = \{z(i,j) : i, j \in N\} \cup \{0\}$, and note that this set can be constructed in $O(n^2)$ by simply enumerating all pairs of products. The candidate assortments can then be captured through the set $\mathcal{Y} = \{\hat{y}(z) : z \in \mathcal{Z}\}$, and based on the discussion above, we know that $y^* = \arg\max_{y \in \mathcal{Y}} R(y)$. From a computational perspective, the most burdensome step is that of computing the set of candidate assortments $\mathcal{Y}$. To see this, note that for each $z \in \mathcal{Z}$, in order to compute the assortment $\hat{y}(z)$ we must compute the relative ordering of the contributions $c_i(z)$ for each product $i \in N$. Rusmevichientong et al. (2010) show that by first sorting $\mathcal{Z}$, these relative orderings can be computed recursively in $O(n^2)$ and hence the total run time for computing $\mathcal{Y}$ in $O(n^2 \log n)$ since the set of intersection points $\mathcal{Z}$ must be sorted. In what follows, we give an algorithm that finds y^* is $O(n^2)$ by never fully computing the set $\mathcal{Y}$. Instead, a bisection approach is used to find the assortment $\hat{y}(\hat{z})$ associated with the fixed point $\hat{z}$.

Our approach begins by computing $\mathcal{Z}$. We then run the bisection algorithm given in Algorithm 1, where the function $Med(S)$ returns the median value of a collection of numbers of S. In Algorithm

1, we maintain throughout that $z_- \leq \hat{z} \leq z_+$. Moreover, when $|\bar{Z}_t| = 2$, we know that the ordering between the contributions $c_i(z)$ does not change for any z satisfying $z_- \leq z \leq z_+$ and hence for any such z we have that $\hat{y}(z) = y^*$. The following Proposition establishes that the running time of Algorithm 1 is indeed $O(n^2)$, where the bottleneck step is in computing $\mathcal{Z}$.

PROPOSITION 1. *The running time of Algorithm 1 is $O(n^2)$*

W e begin by showing that the while loop runs at most $L = O()$ times. To see this, note that we have $\bar{Z}_{t+1} \leq \frac{1}{2}\bar{Z}_t + 1$ due to lines 10 or 13. We can compute $\bar{Z}_{t+1}$ in lines 10 or 13 by simply enumerating over all values $z \in \bar{Z}_t$ and checking for the desired condition on z. The total runninig time of this approach over the L iteration of the algorithm is $\sum_{t=0}^{L} |\bar{Z}_t| = O(n^2)$. Finally, since the median value of $\bar{Z}_t$ can be computed in $O(|\bar{Z}_t|)$, we get a total running time of $\sum_{t=0}^{L} O(|\bar{Z}_t|) = O(n^2)$ for computing the median value z_m over all iterations of the algorithm. Combining all of the steps, the overall run time is $O(n^2)$.

For each arriving customer, we use Algorithm 1 to determine the set of six products to display. Even though the algorithm runs in $O(n^2)$, our implementation on Alibaba easily runs within the 50-millisecond threshold and has never timed-out.

B. Extensions of the MNL Assortment Problem

In this section, we present two extensions of *MNL-Card*, which consider additional operational levers that Alibaba could use to improve revenues. For each such lever, we formulate the new assortment problem and then present a general approach that can be used to either solve the problem optimally, or provide an approximate solution with a provably near optimal performance guarantee. We note that the intent of this section is to present new theoretical results that when applied, have the potential to increase Alibaba's revenue. Unfortunately, the current set-up of our field experiments does not allow us to set prices or change product display icons, and hence these result are theoretical in nature.

1. *Joint pricing and assortment*: In this version of the problem, a retailer must simultaneously decide which products to offer and the prices to charge for each offered product. We assume that each offered product must be priced at one of m prices, indexed by the set $K = \{1, \ldots, m\}$. If the retailer chooses to price product i at price j, then the revenue and MNL-based preference weight of this product are given by r_j and v_{ij} respectively. The preference weight could also be subscripted by the arriving customer type t, however we drop this dependence for ease of exposition. We let $y_{ij} \in \{0,1\}$ be a binary indicator of whether product $i \in N$ is offered at price $j \in K$. In this case, the set of feasible assortment and pricing decisions can be captured through the set

$$\mathcal{F}_1 = \{y \in \{0,1\}^{mn} : \sum_{j=1}^{m} y_{ij} \leq 1 \,\forall\, i \in N\} \cap \{y \in \{0,1\}^{mn} : \sum_{i=1}^{n}\sum_{j=1}^{m} y_{ij} = 6\},$$

which captures the notion that each product can only be offered at one price and that we must continue to offer six product displays. With this notation in hand, we present the corresponding joint pricing and assortment problem below

$$\max_{y \in \mathcal{F}_1} R(y) = \max_{y \in \mathcal{F}_1} \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} r_i y_{ij}}{1 + \sum_{i=1}^{n} \sum_{j=1}^{m} v_{ij} y_{ij}}. \tag{7}$$

2. *Icon display size*: Here, we consider the problem of optimally choosing the size of the icons corresponding to each displayed product. Note that in the current setup, the icons of all six displayed products are the same size (See Figure 2), but in other settings on Alibaba, the icon size may be different, which means that this extension is practically relevant. To model our updated setting in which product icon sizes do not have to be homogeneous, we assume that the icon of each displayed product can take on one of m sizes indexed by the set $K = \{1, \ldots, m\}$, and use c_j to be the screen space consumed by an icon of size $j \in K$. We assume the total available screen space is C. Further, we use v_{ij} to be the MNL preference weight of product i when it is offered with an icon of size j. In this way, we assume that the preference weight of each product is influenced by the size of the icon used to display this product to customers. In this setting, the feasible assortments can be captured through the set

$$\mathcal{F}_2 = \{y \in \{0,1\}^{mn} : \sum_{j=1}^{m} y_{ij} \le 1 \; \forall\, i \in N\} \cap \{y \in \{0,1\}^{mn} : \sum_{i=1}^{n} \sum_{j=1}^{m} c_j y_{ij} \le C\},$$

which reflect the constraints that we must select a single icon size for each displayed product as well as the restriction that we cannot use more than the C units of available screen space. Note that in this case, it is possible for more than six products to be offered. The corresponding assortment problem is then

$$\max_{y \in \mathcal{F}_2} R(y) = \max_{y \in \mathcal{F}_2} \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} r_i y_{ij}}{1 + \sum_{i=1}^{n} \sum_{j=1}^{m} v_{ij} y_{ij}}. \tag{8}$$

For an appropriately chosen constraint matrix A^l and right hand side vector b^l, it is straightforward to see that each collection of feasible assortment $\mathcal{F}_1$ and $\mathcal{F}_2$ can be encoded through the polytopes $A^l y \le b^l$ for $l = 1, 2$. If the matrix A^l is totally unimodular, Davis et al. (2013) show that these assortment problems can be formulated as a tractable linear program, where both the number of variables and constraints grows linearly in the number of products. For example, when a cardinality constraint and a collection of discrete pricing constraints are combined as is done to create $\mathcal{F}_1$, then totally unimodularity is preserved. Hence the following linear program, which was originally presented in in Davis et al. (2013), provides an optimal polynomial time algorithm to solve the joint pricing and assortment problem.

$$\max \; \sum_{i=1}^{n} \sum_{j=1}^{m} r_i w_{ij} \tag{\textit{Pricing LP}}$$

$$w_0 + \sum_{i=1}^{n}\sum_{j=1}^{m} w_j = 1$$

$$\frac{w_{ij}}{v_{ij}} \le w_0 \quad \forall\, j = 1, \ldots, n$$

$$\sum_{j=1}^{m} \frac{w_{ij}}{v_{ij}} \le w_0 \quad \forall\, i = 1, \ldots, n \tag{9}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{m} \frac{w_{ij}}{v_{ij}} = 6 \cdot w_0 \tag{10}$$

$$w_{ij} \ge 0 \;\forall\, j = 1, \ldots, n,$$

where the optimal decision variable w^*_{ij} can be interpreted as the fraction of time that product i is purchased at price j under the optimal assortment. In other words, if y^*_{ij} is the optimal solution to problem (7), then Davis et al. (2013) show that

$$w^*_{ij} = \frac{v_{ij} y^*_{ij}}{1 + \sum_{i=1}^{n}\sum_{j=1}^{m} v_{ij} y^*_{ij}}.$$

To help give some intuition as to why the above linear program captures problem (7), we show that constraints (9) and (10) together encode $\mathcal{F}_1$. To start, note that under the optimal solution w^*_{ij}, constraint (9) can be re-arranged as follows

$$\sum_{j=1}^{m} \frac{w^*_{ij}}{v_{ij}} \cdot \frac{1}{w^*_0} = \sum_{j=1}^{m} y^*_{ij} \le 1,$$

which is exactly encodes the notion that each product can only offered at a single price. Further, constraint (10) can similarly be re-arranged as follows

$$\sum_{i=1}^{n}\sum_{j=1}^{m} \frac{w^*_{ij}}{v_{ij}} \cdot \frac{1}{w^*_0} = \sum_{i=1}^{n}\sum_{j=1}^{m} y^*_{ij} \le 6,$$

which encodes the cardinality constraint.

Unfortunately, however, when a knapsack or space constraint is added as is done to create $\mathcal{F}_2$, totally unimodularity of the constraint matric A^2 no longer holds and hence the linear programming approach of Davis et al. (2013) is seemingly rendered ineffective. Surprisingly, however, we are to salvage the optimal solution to this linear programming to produce a feasible assortment that garners an expected revenue of at least one-third of the optimal expected revenue.

B.1. Icon Display Size Assortment Problem

In this section, we consider the problem Alibaba faces in optimizing the sizes of each display icon for the assortment of products offered to each arriving customer. As discussed in Appendix B, this problem is formally stated as

$$\mathrm{OPT} = \max_{y \in \mathcal{F}_2} \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} r_i y_{ij}}{1 + \sum_{i=1}^{n}\sum_{j=1}^{m} v_{ij} y_{ij}},$$

where we remind the reader that $\mathcal{F}_2 = \{y \in \{0,1\}^{mn} : \sum_{j=1}^m y_{ij} \leq 1 \ \forall \ i \in N\} \cap \{y \in \{0,1\}^{mn} : \sum_{i=1}^n \sum_{j=1}^m c_j y_{ij} \leq C\}$. To view this problem from a more standard lens in which the assortment problem simply consists of choosing a subset of products to offer, we assume the retailer is given m copies of each product $i \in N$ (one for each icon size), and impose the constraint that only one copy of each product can ever be offered. Hence each product can be viewed as a tuple (i,j), where the first entry indicates the item $i \in N$ and the second entry gives the icon size $j \in K$. For notational convenience moving forward, we let $\bar{N} = \{(i,j) : i \in N, j \in K\}$ to be the set containing all such tuples. Further for assortment $S \subseteq \bar{N}$, and product $(i,j) \in \bar{N}$, we let

$$P_{(i,j)}(S) = \frac{v_{ij}}{1 + \sum_{(k,l) \in S} v_{kl}},$$

be the MNL purchase probabilities and $R(S) = \sum_{(i,j) \in S} r_i \cdot P_{(i,j)}(S)$ be the expected revenue of this assortment. We devote the remainder of this section to proving the following theorem.

THEOREM 2. *There is a polynomial time algorithm that produces an assortment $S \subseteq \bar{N}$ that satisfies $R(S) \geq \frac{1}{3} \cdot \mathrm{OPT}$.*

To begin, we note that the matrix A needed that encodes the constraints $\{y \in \{0,1\}^{mn} : \sum_{j=1}^m y_{ij} \leq 1 \ \forall \ i \in N\}$ is an interval matrix, meaning the columns can be organized so each row is a collection of contiguous ones. Since interval matrices are totally unimodular, we can use the following linear programming formulation (Davis et al. 2013) of the icon display assortment problem, in which we do not yet impose the constraint limiting the screen space consumption of the offered assortment.

$$\max \sum_{i=1}^n \sum_{j=1}^m r_i w_{ij} \tag{Icon LP}$$

$$w_0 + \sum_{i=1}^n \sum_{j=1}^m w_j = 1 \tag{11}$$

$$\frac{w_{ij}}{v_{ij}} \leq y_0 \quad \forall \, j = 1, \ldots, n \tag{12}$$

$$\sum_{j=1}^m \frac{w_{ij}}{v_{ij}} \leq y_0 \quad \forall \, i = 1, \ldots, n \tag{13}$$

$$w_{ij} \geq 0 \ \forall \, j = 1, \ldots, n. \tag{14}$$

Let w_{ij}^* be the optimal solution to the to *Icon LP*. Davis et al. (2013) shows that the extreme points of this linear program have special structure. Namely, that there is a bijection between extreme points and feasible assortments; for each feasible assortment $S \subseteq \bar{N}$, there is an extreme point of *Icon LP* that satisfies

$$w_{ij} = \begin{cases} \frac{v_{ij}}{1+\sum_{(k,l) \in S} v_{kl}}, & \text{if } (i,j) \in S \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

Hence, there is a positive coordinate for each offered product whose magnitude is exactly the purchase probability under S. Letting $S^* = \{(i,j) \in \bar{N} : w^*_{ij} > 0\}$, we get that $R(S^*) = \sum_{i=1}^n \sum_{j=1}^m r_i w^*_{ij}$ and hence the objective value of *Icon LP* correctly computes the optimal expected revenue of S^*. Further, under the optimal solution w^*_{ij}, constraint (13) becomes

$$\sum_{j=1}^m \frac{w^*_{ij}}{v_{ij}} \cdot \frac{1}{w^*_0} = \sum_{j=1}^m \mathbb{1}_{w^*_{ij}>0} \le 1,$$

which correctly encodes the notion that at most one icon size will be picked for each displayed product under S^*.

Next, we add the following constraint to *Icon LP*

$$\sum_{i=1}^n \sum_{j=1}^m c_j \cdot \frac{w_{ij}}{v_{ij}} \le C \cdot w_0, \tag{16}$$

which would capture the limit on the screen space that the displayed product icons can take up if the optimal decision variables continued to have the structure given in (15). Note that if this were indeed the case, then this constraint becomes

$$\sum_{i=1}^n \sum_{j=1}^m c_j \cdot \frac{w^*_{ij}}{v_{ij}} \cdot \frac{1}{w^*_0} = \sum_{i=1}^n \sum_{j=1}^m c_j \cdot \mathbb{1}_{w^*_{ij}>0} = \sum_{(i,j)\in S^*} c_j \le C,$$

as desired. Unfortunately, adding this constraint breaks the total unimodularity of the constraint structure, since adding a knapsack constraint clearly does not preserve total unimodularity, and hence the structure presented in (15) does not hold for the new optimal solution to the linear program. However, since we have added only a single constraint to *Icon LP*, the new optimal solution can be written as a convex combination of two extreme points of the polytope described by (11)-(14), which both have the structure presented in (15).

More formally, let $\hat{w} = (\hat{w}_{11}, \ldots, \hat{w}_{nm})$ be the optimal solution to *Icon LP* after adding constraint (16). For some $\alpha \in [0,1]$, we must have that $\hat{w} = \alpha \cdot w' + (1-\alpha) \cdot w''$, where $w' = (w'_{11}, \ldots, w'_{nm})$ and $w'' = (w''_{11}, \ldots, w''_{nm})$ are extreme points of the polyhedron described by the constraints (11)-(14). Further we let $S' = \{(i,j) \in \bar{N} : w'_{ij} > 0\}$ and $S'' = \{(i,j) \in \bar{N} : w''_{ij} > 0\}$ be the assortments related to these two extreme points.

Moving forward, we let

$$\mathcal{R}^* = \sum_{i=1}^n \sum_{j=1}^m r_i \hat{w}_{ij}$$
$$\mathcal{R}' = \sum_{i=1}^n \sum_{j=1}^m r_i w'_{ij} = R(S')$$
$$\mathcal{R}'' = \sum_{i=1}^n \sum_{j=1}^m r_i w''_{ij} = R(S''),$$

where $\mathcal{R}'$ and $\mathcal{R}''$ give the expected revenues of the assortments S' and S'' due to the structure of any extreme point given in (15). Further, note that we have $\mathcal{R}^* = \alpha \cdot \mathcal{R}' + (1-\alpha) \cdot \mathcal{R}'' \geq \text{OPT}$, where the inequality follows since we have established that *Icon LP* with constraint (16) is a relaxation of the original problem. Further, we let

$$\mathcal{C}^* = \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{w_{ij}^*}{v_{ij}} \cdot \frac{1}{w_0^*}$$
$$\mathcal{C}' = \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{w_{ij}'}{v_{ij}} \cdot \frac{1}{w_0'} = \sum_{(i,j) \in S'} c_j$$
$$\mathcal{C}'' = \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{w_{ij}''}{v_{ij}} \cdot \frac{1}{w_0''} = \sum_{(i,j) \in S''} c_j,$$

where for the latter two expressions we use the structure of any extreme point that is given in (15). Further, we have that $\mathcal{C}^* = \alpha \cdot \mathcal{C}' + (1-\alpha) \cdot \mathcal{C}'' \leq C$, which results by simply plugging $\hat{w}$ into constraint (16).

Next, we assume without loss of generality that $\mathcal{C}' \geq \mathcal{C}''$ and hence the assortment S'' is feasible since we must have $\mathcal{C}'' \leq C$. Thus, we have found one initial feasible solution with expected revenue $\mathcal{R}''$. Next, we show how to find an assortment $\hat{S}$ that satisfies $R(\hat{S}) \geq \alpha \cdot \frac{\mathcal{R}'}{2}$. Consider the following linear programming relaxation of a constructed knapsack problem in which the items are the products of S'.

$$Z_{knap}^* = \max \sum_{(i,j) \in S'} r_j P_{(i,j)}(S') x_{ij} \qquad \text{(KNAP)}$$
$$\text{s.t.} \sum_{(i,j) \in S'} c_j x_{ij} \leq C$$
$$0 \leq x_{ij} \leq 1.$$

In this knapsack problem, the value of item $(i,j) \in S'$ is $r_j P_{(i,j)}(S')$ and its space consumption is c_j. Note that the solution $\tilde{x}_{ij} = \alpha$ for all $(i,j) \in S'$ is a feasible solution to the knapsack linear program and has objective function value equal to $\alpha \cdot \mathcal{R}'$. Hence the optimal objective function value of the knapsack linear program is at least $\alpha \cdot \mathcal{R}'$. Further, it is well known that the optimal solution to the linear programming relaxation of any knapsack problem has at most one fractional variable value. Let x^* be an optimal solution to the linear programming relaxation, and let $\hat{S}_1 = \{(i,j) \in S' : x_{ij}^* = 1\}$ and $\hat{S}_2 = \{(i,j) \in S' : 0 < x_{ij}^* < 1\}$. Note that $|\hat{S}_2| \leq 1$. The following lemma bounds the revenue of the best of these two solutions.

Lemma 1. *Let* $\hat{S} = \arg\max_{S \in \{\hat{S}_1, \hat{S}_2\}} R(S)$. *Then,* $\sum_{(i,j) \in \hat{S}} c_j \leq C$ *and*

$$R(\hat{S}) \geq \frac{\alpha}{2} \cdot \mathcal{R}'.$$

By the feasibility of x^*, we must have that $\sum_{(i,j)\in\hat{S}_1} c_j \leq C$. Further, without loss of generality, every product has a space consumption of at most C. Therefore, $\sum_{(i,j)\in\hat{S}_2} c_j \leq C$ as well, and so $\sum_{(i,j)\in\hat{S}} c_j \leq C$. Next, we consider the revenue of assortment $\hat{S}$.

$$\begin{aligned}
R(\hat{S}) &= \max\left\{R(\hat{S}_1), R(\hat{S}_2)\right\} \\
&= \max\left\{\sum_{(i,j)\in\hat{S}_1} r_j P_{(i,j)}(\hat{S}_1), \sum_{(i,j)\in\hat{S}_2} r_j P_{(i,j)}(\hat{S}_2)\right\} \\
&\geq \max\left\{\sum_{(i,j)\in\hat{S}_1} r_j P_{(i,j)}(S'), \sum_{(i,j)\in\hat{S}_2} r_j P_{(i,j)}(S')\right\} \\
&\geq \frac{1}{2}\left\{\sum_{(i,j)\in S'} r_j P_{(i,j)}(S') x^*_{(i,j)}\right\} \\
&\geq \frac{\alpha}{2}\cdot\mathcal{R}'.
\end{aligned}$$

The first inequality comes from the fact that $\hat{S}_1, \hat{S}_2 \subseteq S'$, and hence the MNL purchase probabilities of each product can only decrease moving from $\hat{S}_1$ or $\hat{S}_2$ to S' . The second inequality results from the fact that

$$\sum_{(i,j)\in\hat{S}_1} r_j P_{(i,j)}(S') + \sum_{(i,j)\in\hat{S}_2} r_j P_{(i,j)}(S') \geq \sum_{(i,j)\in S'} r_j P_{(i,j)}(S') x^*_{ij}.$$

The last inequality results since $Z^*_{knap} = \sum_{(i,j)\in S'} r_j P_{(i,j)}(S') x^*_{(i,j)} \geq \alpha\cdot\mathcal{R}'$

At this point, we consider either offering the assortment S'' or $\hat{S}$. Let $\bar{S} = \arg\max_{S\in\{S'',\hat{S}\}} R(S)$. To conclude the proof of Theorem 2 we show that $R(\bar{S}) \geq \frac{1}{3}\cdot\text{OPT}$. To do so, note that we have that $R(\bar{S}) = \max\{\mathcal{R}'', R(\hat{S})\} \geq \max\{(1-\alpha)\cdot\mathcal{R}', \frac{\alpha}{2}\cdot\mathcal{R}'\}$. Recall that $\alpha\cdot\mathcal{R}' + (1-\alpha)\cdot\mathcal{R}'' \geq \mathcal{R}^*$. Therefore, if $(1-\alpha)\cdot\mathcal{R}'' \geq \frac{1}{3}\cdot\mathcal{R}^* \geq \frac{1}{3}\cdot\text{OPT}$, the result holds. Otherwise, $\alpha\cdot\mathcal{R}' \geq \frac{2}{3}\cdot\mathcal{R}^*$, and again, the result holds.

C. Implementation Details for Estimation Case Study

In this section, we provide the code we use for fitting the MNL and machine learning models within the case study presented in Section 3.3.

C.1. Tensorflow MNL MLE implementation

This section contains a Tensorflow python implementation of the MNL MLE problem given in (1). The input parameters have the following meaning, given that training data set contains records of τ customer arrivals.

- num_features: number of features in the data set. In our case this is 25.
- assort_size: The number of products in each offered assortment. In our case this is 6.

- offer_data: This is a assort_size $\cdot \tau \times$ num_features numpy array, where each row is a feature vector X_{jt}.
- purchase_data: This is a assort_size $\cdot \tau \times 1$ numpy array, where each row is a binary indicator of whether the given product was purchased by the arriving customer.
- batch_size: The number of data points to use for training in each iteration of stochastic gradient descent. We use 20,000.
- learning_rate: The learning rate of the stochastic gradient descent algorithm. We use 0.05.

```
def Estimate_MNL_Classic(num_features, assort_size, offer_data, purchase_data, batch_size, learning_rate):

    #Placeholders for offer data
    offer_mnl = tf.placeholder(tf.float32, [None, assort_size, num_features])

    #Placeholder for purchase data
    purchase_mnl = tf.placeholder(tf.float32, [None, num_features])

    #Create the variables to be estimated - the MNL feature weights
    W = tf.Variable(tf.random_normal(shape=[num_features], mean=0, stddev=0.01), name="weights")

    #Compute the log likelihood
    first_term = tf.reduce_sum(tf.multiply(purchase_mnl, W),1)
    second_term =tf.log(tf.reduce_sum( tf.exp(tf.reduce_sum(tf.multiply(offer_mnl,W),2)), 1)  + 1)
    cost = tf.reduce_sum(second_term-first_term)

    #Optimization
    optimizer = tf.train.AdamOptimizer(learning_rate).minimize(cost)
    init = tf.global_variables_initializer()

    epoch_count= 0
    with tf.Session() as sess:
        sess.run(init)
        #Stopping criterion
        while epoch_count<2000:
            num_batches = int(purchase_data.shape[0]/batch_size)
            log_like = 0
            for b in range(num_batches+1):
                if b<num_batches:
                    purchase_batch = purchase_data[b*batch_size:(b+1)*batch_size,:]
                    offer_batch = offer_data[b*batch_size:(b+1)*batch_size,:,:]
                else:
                    purchase_batch = purchase_data[b*batch_size:,:]
```

```
                offer_batch = offer_data[b*batch_size:,:,:]

            num_rows = purchase_batch.shape[0]
            _,neg_log_like = sess.run(fetches=[optimizer, cost], \
                                  feed_dict={offer_mnl:offer_batch, purchase_mnl:purchase_batch})
            log_like+=neg_log_like

        epoch_count+=1

    #Get final weights
    return sess.run(W)
```

C.2. Fitting the Machine Learning Models

This section contains a python implementation of Catboost for estimating click and purchase probabilities, where we have done no hyperparameter tuning. The input parameters have the following meaning.

- df_train: Pandas dataframe containg the training data, which has a row for each product displayed to each each customer and whose columns give the various feature values. Further, there are additional binary columns (is_click and is_buy) that indicate whether the given product was clicked and/or purchased.
- feature_list: A list of column names corresponding to the features that will be used to fit the Catboost model.

```
def Estimate_ML_Catboost(df_train, feature_list):

    #Only look at clicked products
    clicked_train = df_train.loc[df_train.is_click ==1 ,:]

    #Training data for estimating conditional purchase probs
    X_train_buy = np.array(clicked_train.loc[: , feature_list])
    y_train_buy = np.array(clicked_train.is_buy)

    #Training data for estimating click probs
    X_train_click = np.array(df_train.loc[: , feature_list])
    y_train_click  = np.array(df_train.loc[: , "is_click"])

    #Fitting the two catboost models
    model_buy = CatBoostClassifier().fit(X_train_buy, y_train_buy)
    model_click = CatBoostClassifier().fit(X_train_click, y_train_click)

    return model_buy, model_click
```

D. The Comparison of Full-feature MNL and Machine Learning Model

Table 8 All-feature Model Financial Performance

	AF-MNL	AF-ML
RevenuePerVisit (RMB)	4.79	4.64
Difference (All p-values)	0.15	
Relative Improvement	3.37%	
T-test p-value	< 0.0001	
Observations	3,152,580	3,148,217

Notes. The table reports the average financial performance, in terms of revenue per customer visit, across different algorithms during our five-day-long experimental period (September 20, 2018 - September 24, 2018).

We finished implementing and testing our MNL-based approach on all features by September 15th, 2018. Therefore, we conducted a five-day-long experiment from September 20nd, 2018 to September 24th, 2018 where customers are randomly assigned into the *the all-feature-MNL-choice-model-based approach (AF-MNL approach)* and *the all-feature-ML-based approach (AF-ML approach)* based on a unique hash number derived from the given customer's ID and an experiment ID. The AF-ML approach is exactly the same as the all-feature approach in Section 6 except that the training data is in August and September instead of February and March. Similarly, the AF-MNL approach is similar to the MNL-based approach in Section 6 except (a) the training data has advanced to August and September; and (b) the estimation process uses all features instead of the top 25 features.

Over the five days of our experiment, we observe $3,591,021$ customer arrivals from $2,247,663$ million unique customers. $1,125,381$ of these customers are randomly assigned to the MNL-choice-model-based approach on all features (i.e, AF-MNL approach) while $1,122,282$ are assigned to Alibaba's original machine learning approach on all features (i.e, AF-ML approach). The customers under AF-MNL approach collectively spend $15,114,748$ RMB during the five days, while the customers in the AF-ML approach spend $14,621,580$ RMB, an improvement of $493,168$ RMB (i.e., 3.37% during the experimental period.

Table 8 presents the GMV per customer visit generated by these two approaches on all features. The first row shows that the AF-MNL and AF-ML approaches generate RMB 4.79 and 4.64 per customer visit respectively, and the different is 0.15 RMB per customer visit; in other words, the AF-MNL approach improves the revenue per customer visit by 3.37% compared to the AF-ML approach (p-value < 0.0001). This demonstrate that the MNL-based approach out-performs the machine-learning-based approach even if both approaches are utilizing all features. We note that the machine-learning-based approach on all features is exactly the algorithm and the feature set that is used by Alibaba to recommend products prior to our collaboration. This shows that our algorithm improves the state-of-art recommender system of Alibaba by 3.37%, which leads to the adoption of our algorithm as the main recommendation algorithm in this setting on Alibaba. We also note that this improvement based on all features (i.e., 3.37%) is more modest than the improvement based on only top 25 features (i.e., 28.0%), which may demonstrate that machine-learning-based approaches can more easily scale up to more features.