Dear Prize Committee,

Dennis Zhang and I would like to submit our paper titled “Taking Assortment Optimization from Theory to Practice: Evidence from Large Scale Field Experiments at Alibaba” to the 2018 Olin Award competition. This paper is currently under review at Management Science.

One of the most fundamental operational problems faced by any retailer is that of deciding which set of products to make available for purchase to each arriving customer. In the growing world of e-commerce, the set of products displayed to each customer can be highly personalized based on the historical purchasing behavior of the customer at hand. In general, this product display problem is addressed by first estimating the demand for each product, and then using these estimates to seed an optimization problem, whose optimal solution determines the set of products that will be offered.

In this paper, we compare the performance of two approaches for finding the optimal set of products to display to customers landing on Alibaba, the world’s largest retail platform. 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 hundreds 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 considers the product display problem from a novel perspective: instead of using powerful machine learning tools for prediction, we instead opt to capture demand through a simple multinomial logit (MNL) choice model. While the estimation step in this approach is simpler, the resulting optimization problem used to determine the set of products to display is more nuanced because under the MNL choice model, the purchase probabilities for each product depend on the set of products displayed. This structure renders simple sorting algorithms ineffective, and hence leads us to develop a novel algorithm, which we show can be implemented quite efficiently.

The ever-growing suite of machine learning algorithms are powerful tools for prediction; they help 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. As a consequence, the efficacy of any subsequent optimization that is seeded with these estimates and used to make critical operational decisions can be dramatically lowered. In contrast, the MNL model is simple, but it was created to capture
customer purchasing behavior and, more specifically, substitution patterns. Hence because our MNL-based approach uses a model catered to the specific business setting that we consider, we ultimately seem to solve a more sophisticated optimization problem. Our experiments show that our MNL-based approach generates 28% higher revenue per visit compared to the current machine learning algorithm with the same set of features. Given that the weekly sales in this experimental setting is 21.6 million dollars, 28% improvement can be translated into 6.05 million dollars per week.
Taking Assortment Optimization from Theory to Practice: Evidence from Large Field Experiments 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 hundreds 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 \times 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 28% 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. In addition to developing the first full-scale, choice-model-based product recommendation system, we also shed light on new directions for improving such systems for future use.

**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

*The first two authors contributed equally and are ranked alphabetically.
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 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 three products at most 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 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. **The Estimation Problem:** Can a choice model’s parameters be efficiently estimated from historical data? Moreover, are these estimates accurate and will operational decisions made based on these estimates be profitable?

2. **The Assortment Problem:** 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, to the best of our knowledge, this sort of implementation has never been carried
out in practice, and hence the efficacy of assortment optimization remains mainly theoretical in nature.

This work aims to bridge the gap between theory and practice as it relates to assortment optimization. To do so, we provide the first large-scale implementation of a revenue management system that uses assortment optimization to guide product display decisions. We develop our algorithm as well as the corresponding engineering system and conduct our large-scale field experiment in collaboration with the Alibaba Group, a Chinese online and mobile commerce company whose Gross Merchandise Value (GMV) has surpassed US$485 billion as of 2016. This makes Alibaba the world’s largest retailer, overtaking Walmart, which posted revenues of $482.1 billion for the same year. We consider a setting where Alibaba must present customized six-product assortments to inquiring customers, with the goal of maximizing revenue. Henceforth, we refer to this problem as the Alibaba Product Display Problem. Our newly built system uses a collection of product features and real-time customer features as input to a cardinality-constrained assortment problem, which is solved online to determine the optimal set of products to be displayed to the given customer at hand. In Section 2, we provide a detailed overview of Alibaba’s product display problem that is the canvas for our field experiments. Section 3 includes a summary of the machine-learning approach that is the current practice on Alibaba and which acts as a benchmark for our choice-modeling-based approach.

The implementation of this revenue management system unfolds in two steps that sequentially address the estimation and assortment problems described above as they relate to the setting we study at Alibaba. As a precursor, however, we first must select an underlying customer choice model to be the presumed mechanism governing customer purchasing behavior. We choose to use the classic multinomial logit (MNL) model, which was pioneered by Luce (1959) and McFadden (1974) and later shown to be able to capture a diverse array of customer purchasing patterns (see Vulcano et al. (2012)) while also admitting a tractable cardinality-constrained assortment problem (see Rusmevichientong et al. (2010) and Davis et al. (2013)). We defer a formal description of the MNL choice model to Section 4, but we note here that another intriguing feature of this model is that the deterministic component of its utility can be written as a linear combination of product and customer features. This parameterization allows for a more concise representation of the model, which in turn simplifies the estimation step since fewer parameters need to be estimated. Further, since customer features can be easily incorporated within this parameterization, the model can be

---

textunderscore pdf/p160505.pdf

2 https://www.forbes.com/sites/jlim/2016/05/05/alibaba-fy2016-revenue-jumps-33-ebitda-up-28/#5f15a05f53b2
shaped around the unique attributes and past purchasing/click behavior of each arriving customer, allowing us to capture a fine-grained picture of customer purchasing patterns.

As mentioned above, the first task in building our revenue management system is to fit a featurized multinomial choice model to historical sales data. The prevailing method in the literature for accomplishing this task is through approaches based on maximum likelihood estimation (MLE) (see Vulcano et al. (2012), van Ryzin and Vulcano (2017), and Topaloglu and Simsek (2017)). We follow suit and consequently fit our featurized multinomial logit choice model by directly maximizing the log-likelihood, which McFadden (1974) shows to be concave in the model parameters. However, due to the sheer volume of visit data generated each day on Alibaba, we cannot simply solve the resulting MLE problem using an off-the-shelf nonlinear optimization package. Instead, we implement a parallelized version of stochastic gradient descent through TensorFlow, which we show is capable of solving MLE problems using data from tens of millions of past customers.

Once the multinomial choice model has been estimated, we move on to the cardinality-constrained assortment problem, which is used to determine the customized six-product displays. Since this assortment problem is solved online for each arriving customer, efficiency is at a premium. In fact, the upper limit on run time to avoid page delays, as supplied by Alibaba, is 50 milliseconds for solving the assortment problem. In the existing literature, two approaches have been developed for solving cardinality-constrained assortment problems under the multinomial logit choice model. The first optimal polynomial-time approximation scheme is due to Rusmevichientong et al. (2010), who provide a purely combinatorial algorithm that cleverly exploits the structure of the multinomial logit model. The run time of this approach is $O(n^2 \log(n^2))$, where $n$ is the number of products that potentially could be included in the offered assortment. Davis et al. (2013) later provide a linear programming formulation of the problem in which the number of both decision variables and constraints grow linearly in $n$. Despite the concise nature of this formulation, we elect to use the combinatorial algorithm of Rusmevichientong et al. (2010) due to the fact it can be fully implemented more easily in a low-level programming language, such as C++. In Section 4, we provide a novel implementation of this approach that runs in $O(n^2)$. Given the 50-millisecond threshold run time and that $n$ can be in the thousands, this $O(\log(n^2))$ decrease in run time due to our novel algorithm is a nontrivial improvement.

While the novel implementation ideas described above are critical for developing a practical revenue management system, they ultimately are rendered useless if the system does not produce profitable product displays. In this regard, our foremost contribution is that we show that product recommendation systems built on the framework of assortment optimization have the potential to outperform machine-learning-based approaches. We show this result by conducting a large-scale field experiment for one week involving more than 5 million customers, which compares the
performance of our MNL-based approach with Alibaba’s state-of-the-art machine-learning-based approach. Our MNL model is fit to a rolling week of past historical sales data using a collection of the top 25 most descriptive customer and product features. We then use this estimated model to seed a cardinality-constrained assortment problem that determines the six-product displays. We find our MNL-based approach generates 28% higher revenue per visit compared to the machine-learning-based approach that uses the same set of 25 customer and product features. Surprisingly, our MNL-based approach also slightly outperforms Alibaba’s current practice, which embeds hundreds of features within the machine-learning-based approach. We find that the MNL-based approach’s superior performance can be attributed mainly to the fact that it chooses six-product displays that attract customers to more expensive products. However, this superior performance is not observed across all sellers, and hence in Section 6.4 we provide heterogeneous treatment analyses to demonstrate when our MNL-based approach does not perform well compared to the machine-learning-based approach. The full details and results of our experiment are given in Sections 5 and 6.

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.

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.

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 admits 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 a 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.

Retailing operations. Last, 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.

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.

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.\(^3\)

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.”

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

\(^3\) http://about.tmall.com/tmall/fee_schedule
Figure 1 Recommendation on Alibaba in the Public (left) and Private (right) Domains

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 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.
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.\(^4\)

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:

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

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 Machine-Learning-Based Product Display System

In this section, we provide a high-level description of the approach Alibaba currently uses to solve the Alibaba Product Display Problem. For this approach, the purchase probabilities $P_{jt}$ are estimated using machine learning. In what follows, we discuss the various machine learning algorithms that are employed and also provide a description of the most important customer and product features that guide the predictions in this case. We are not able to provide the exact details of the algorithms due to security reasons; however, we intend to provide enough details so that the advantages and drawbacks of this approach can be well understood.

Available product and customer features. We begin by discussing the makeup of the available historical sales data used to fit the machine learning algorithms. This training data is composed of historical sales information from $\tau$ 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 picture 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 used include demographic information, such as age, gender, and registration time. Registration time is included since customers who registered early on the platform tend to have different spending habits compared to customers who registered later. This is likely a result of Alibaba’s expansion strategy. In the last 10 years, Alibaba has expanded its services from first-tier cities to second- and third-tier cities. 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.5

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 used within the machine learning estimation schemes that are implemented.

The machine learning estimation problem. In what follows, we formalize the estimation problem whose solution gives the estimates of $P_{jt}$ that are used within the machine-learning-based product display system. 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 $X_{jt} = [x_j, x_t, x_{jt}]$ gives the features associated with the particular observation, which can be partitioned into three sub-vectors of features. More specifically, we let $x_j$ be the product-specific features, $x_t$ be the customer-specific features, and $x_{jt}$ be joint features that are specific to each customer and product pair. 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 $T = 6T$ (since each customer is shown six products) observations, which we represent as $\text{PurchaseHistory}_{ML} = \{(X_{jt}, C_{jt}, Z_{jt}) : t = 1, \ldots, T, j \in S_t\}$.

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

5 “Collecting” a product on Alibaba is analogous to adding a product to a wish list on Amazon.
estimates of the click probabilities $P(C_{jt} = 1)$, 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 $P(Z_{jt} = 1|C_{jt} = 1)$. 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-boosting 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 contributes the most to the final prediction outcome due to its superior prediction performance compared to that of tree-based models or deep neural nets. 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.

**Solving the Alibaba Product Display Problem.** With purchase probability estimates $P_{jt}(X_{jt})$ available for any customer $t$ and product $j$, we are ready to give our simple algorithm for solving the *Alibaba Product Display Problem*. 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. The reason this algorithm is trivially optimal is 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 well known that the aforementioned greedy algorithm is optimal.

### 3.1. Advantages and Drawbacks of the Machine-Learning-Based Approach

In measuring the efficacy of any approach for the product display problem that we consider on Alibaba, we must consider both the accuracy of its estimates for the purchase probabilities and
the ease with which efficient algorithms can be developed to determine the optimal six-product displays. In what follows, we provide a careful assessment of the upside and downside of using a machine learning approach for the estimation problem and how this choice trickles down to affect the accompanying assortment problem. We begin by emphasizing that for this approach, each observation in PurchaseHistory_{ML}, i.e., \((X_{jt}, C_{jt}, Z_{jt})\) 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 purchase probabilities are \textit{independent} of the assortment of products displayed. While the optimality of the simple greedy algorithm for the assortment component of the \textit{Alibaba Product Display Problem} relies on this structure, what is gained in efficiency potentially is lost when suboptimal 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.

4. The Multinomial-Logit-Choice-Model-Based Product Display System

In this section, we detail our MNL-choice-model-based approach for the \textit{Alibaba Product Display Problem}. We detail this approach by again breaking down its implementation into two steps that sequentially address the estimation and assortment problems. For the estimation problem, we are faced with the task of estimating the parameters of an MNL choice model to tens of millions of historical data points. The scale of the problem not only makes implementing a computationally efficient estimation procedure difficult, but it also renders ineffective past techniques that handle inherent data censorship issues. In what follows, we formalize the estimation problem that we study and provide a detailed description of the aforementioned difficulties that arise, as well our approaches for sidestepping these issues.

The fitted parameters of the underlying MNL model are then used to derive estimates of the purchase probabilities that seed the \textit{Alibaba Product Display Problem}. The resulting assortment problem is the classic 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 run time we are able to improve by a factor of \(O(\log(n^2))\). We provide details of this improved implementation later in this section.
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 $\epsilon_t$. More formally, we have that

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

In order to incorporate product and customer features within the above utility function, one can write the deterministic component of the utility as $V_{jt} = \beta_X X_{jt}$, 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. Among these top 25 features is the propensity score computed from the collaborative filtering techniques. The one-hot encoding of product ID is used within the full-featured machine-learning-based approach but is left out as a feature within the MNL-choice-model-based approach.

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 $V_0 = 0$, which is an assumption that can be made without loss of generality. Under the MNL model, if the retailer offers assortment $S_t \subseteq \mathcal{N}$ to customer $t$, then the probability that product $j \in S_t$ is purchased is given by

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

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 $\beta$ will be fixed and we will define $v_{jt} = e^{\beta_X X_{jt}}$ 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 $\beta$ coefficients. We formulate the likelihood using historical sales data from $\tau$ customers. More specifically, we represent the past purchasing history of the $\tau$ customers as the set
PurchaseHistory\textsubscript{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 \(X_t = \{\tilde{X}_jt : j \in S_t\}\) 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

\[
\max_\beta \mathcal{L}(\beta \mid \text{PurchaseHistory}_{MNL})
\]

where

\[
\mathcal{L}(\beta \mid \text{PurchaseHistory}_{MNL}) = \sum_{t=1}^{\tau} \beta' x_{z_t,t} - \log(1 + \sum_{j \in S_t} e^{\beta' x_{j,t}}).
\]

In problem (1), the objective is the log-likelihood written as a function of the purchasing history of the \(\tau\) customers. In the above MLE problem, we seek the \(\beta\) coefficients, which maximize this log-likelihood function. It is well known (see McFadden (1974)) that the objective function in (1) is concave in the \(\beta\) coefficient. Hence, when \(\tau\) is relatively small, off-the-shelf nonlinear optimization solvers, such as MATLAB’s \textit{fmincon}, 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 \(\tau\) 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 \(\beta\) values. In our setting, 97-99\% 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 \(\beta\) values by a constant \(\delta\).

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 \(\delta\) values, we find that setting \(\delta = 2000\) is the best scaling coefficient.\(^6\)

\(^6\) 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 \(\delta\). The largest performance difference in terms of average revenue per visit between MNL approaches with different values of \(\delta\) is less than 5%.
Solving the Alibaba Product Display Problem. 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 $N = \{1, \ldots, n\}$, 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 $v_{jt} = \beta^* \hat{X}_{jt}$, where $\beta^*$ is the optimal solution to problem (1), after being scaled down by $\delta$. 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:

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

We let $y^*$ be the optimal solution to problem $MNL\text{-}Card$. The first optimal polynomial-time algorithm for problem $MNL\text{-}Card$ is due to Rusmevichientong et al. (2010). They provide a purely combinatorial approach whose run time is $O(n^2 \log(n^2))$. In what follows, we give a novel implementation of this algorithm, which improves upon this previous run time by a factor of $O(\log(n^2))$.

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

$$f(z) = \max_{y \in \mathcal{F}} \sum_{j \in N} v_j (r_j - z) y_j. \quad (2)$$

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 (2). 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 (2). For completeness, we include a proof of the result in Appendix A.

**Theorem 1.** Let $\hat{z} \geq 0$ satisfy $f(\hat{z}) = \hat{z}$, then we have that $R(\hat{y}(\hat{z})) = R(y^*)$. 
In short, Theorem 1 states that if we can find a \( \hat{z} \geq 0 \) that is a fixed point of (2), 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 \( f(z) \) trivially decrease 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

1: \( t = 0 \)
2: \( z_+ \leftarrow 0 \)
3: \( z_- \leftarrow \max_{i \in N} r_i \)
4: \( \bar{Z}_t \leftarrow Z \)
5: \textbf{while} \( \bar{Z}_t > 2 \) \textbf{do}
6: \( z_m \leftarrow Med(\bar{Z}_t) \)
7: \( f \leftarrow f(z_m) \)
8: \textbf{if} \( f < z \) \textbf{then}
9: \( z_- \leftarrow z_m \)
10: \( \bar{Z}_{t+1} \leftarrow \{ z \in \bar{Z}_t : z \geq z_m \} \).
11: \textbf{else}
12: \( z_+ \leftarrow z_m \)
13: \( \bar{Z}_{t+1} \leftarrow \{ z \in \bar{Z}_t : z < z_m \} \)
14: \textbf{end if}
15: \( t \leftarrow t + 1 \)
16: \textbf{end while}
17: \textbf{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, $z(i, j) = \frac{v_i - v_j}{v_i - v_j}$. We denote the set of all such intersection points as $Z = \{z(i, j) : i, j \in N\} \cup \{0\}$, 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 $\mathcal{Y} = \{\hat{y}(z) : z \in Z\}$, 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 $z \in Z$, 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 $Z$, these relative orderings can be computed recursively in $O(n^2)$ and hence the total run time for computing $\mathcal{Y}$ is $O(n^2 \log(n^2))$ since the set of intersection points $Z$ 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 $Z$. 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 $|\tilde{Z}_t| = 2$, 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 proves that the run time of Algorithm 1 is indeed $O(n^2)$, where the bottleneck step is in computing $Z$.

**Proposition 1.** The run time of Algorithm 1 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.

### 4.1. Advantages and Drawbacks of the MNL-Choice-Model-Based Approach

The main advantage of using the MNL choice model to capture customer purchasing patterns is that our estimates of the purchase probabilities now incorporate elements of customer substitution behavior and hence incorporate cannibalization effects that exist, for example, when two similar products are displayed together. While incorporating this realistic element of purchasing behavior doesn’t necessarily guarantee we will end up with more accurate estimates for the purchase probabilities, it is likely that the resulting *Alibaba Product Display Problem* is better able to capture the nuances and difficulties in choosing the optimal set of six products to display to each customer. This idea is perfectly captured in the following example, in which we assume that the machine-learning-based approach has perfectly estimated the purchase probabilities, but because these estimates fail to incorporate substitution behavior, a suboptimal assortment ultimately is
recommended. On the other hand, the MNL-choice-model-based approach does not need to have perfect estimates of the purchase probabilities to recover the optimal assortment due to the fact that we solve a more sophisticated version of the Alibaba Product Display Problem.

Example. Consider a simplified setting of our problem in which \( N = \{1, 2, 3\} \) and the retailer must display exactly two products. The revenues of the three products are \( r_1 = 100, r_2 = 5, r_3 = 5 \). Each arriving customer is assumed to make purchasing decisions according to the same MNL choice model, whose preference weights are \( v_1 = 0.1, v_2 = 0.5, v_3 = 0.1 \). Historically, we assume that the retailer has only offered assortment \( S_1 = \{1, 2\} \) and \( S_2 = \{1, 3\} \) to arriving customers and that each of these assortments appears an equal number of times. Let \( p_i \) be the probability that product \( i \) is purchased. For product 1 we have that

\[
p_1 = \frac{0.1}{1 + 0.1 + 0.5} \times 0.5 + \frac{0.1}{1 + 0.1 + 0.1} \times 0.5 = 7.29%.
\]

Similarly, the marginal purchasing probabilities of products 2 and 3 are

\[
p_2 = \frac{0.5}{1 + 0.1 + 0.5} \times 0.5 + \frac{0.5}{1 + 0.5 + 0.1} \times 0.5 = 31.25%
\]

\[
p_3 = \frac{0.1}{1 + 0.1 + 0.5} \times 0.5 + \frac{0.1}{1 + 0.1 + 0.1} \times 0.5 = 7.29%.
\]

Assume that we have enough data so that the empirical purchase probabilities match these true marginal purchase probabilities. Further, assume that the machine learning approach has perfect estimates of these purchase probabilities. In other words, for each customer \( t \), we assume that the machine learning approach produces estimates of the purchase probabilities \( P_{jt} \) such that \( P_{jt} = p_j \). We let \( S_{ML} \subseteq N \) be the set of products displayed by the machine-learning-based approach. Note that since the machine learning approach ranks the products based on the term \( P_{jt} \cdot r_j \), we will have that \( S_{ML} = \{1, 2\} \). The expected revenue from this offering is

\[
R(S_{ML}) = \frac{0.1 \times 100 + 0.5 \times 5}{1 + 0.1 + 0.5} = 7.8125.
\]

Next, we move on to the MNL-choice-model-based approach, where we assume that the preference weights have been misestimated. More specifically, assume the preference weights estimated from the MNL model are given by \( \{\tilde{v}_1, \tilde{v}_2, \tilde{v}_3\} = \{0.06, 0.1, 0.1\} \), where we have that \( \tilde{v}_1 \neq v_1 \). We let \( S_{MNL} \) be the optimal assortment under the estimated MNL choice model. In this case, we have that \( S_{MNL} = \{1, 3\} \), which garners an expected revenue of

\[
R(S_{MNL}) = \frac{0.1 \times 100 + 0.1 \times 5}{1 + 0.1 + 0.1} = 8.75,
\]

which is over 10% larger than revenue from the perfectly estimated ML model.
This example demonstrates the downside of not factoring in substitution behavior within the underlying demand model. The machine learning model is able to perfectly recover the purchase probabilities, but a suboptimal assortment ultimately is recommended because substitution effects are not accounted for. In fact, the MNL-choice-model-based approach recovers the optimal assortment even when we have $\tilde{v}_1 = \frac{1}{19}$, and hence the estimate of the first product’s preference weight is off by 47%.

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). Our results do not change qualitatively if we use the second week of data. After observing the results of our experiment, Alibaba chose to continue to use our MNL-choice-model-based approach to serve more than 70% of traffic on the coupon sub-pages.

Throughout the experiment, we test the following three approaches:

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 that is described in Section 4. As previously mentioned, 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 3, 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.
Table 1  Summary Statistics

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

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.

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. 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.

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 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.)

7 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.

Electronic copy available at: https://ssrn.com/abstract=3232059
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 1 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 1 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, as the example in Section 4 demonstrates, 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 \( k \).
### Table 2 Model Financial Performance

<table>
<thead>
<tr>
<th>Panel A: Summary Statistics of Financial Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNL    SF-ML    MNL    AF-ML</td>
</tr>
<tr>
<td>RevenuePerVisit (RMB)  5.17    4.04    5.17    5.16</td>
</tr>
<tr>
<td>Difference (All p-values) 1.13 (&lt;0.0001) 0.01 (0.8346)</td>
</tr>
<tr>
<td>Relative Improvement 28.0% 0.2%</td>
</tr>
<tr>
<td>Observations 3,469,129 3,484,555 3,469,129 3,467,965</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Panel B: OLS Regression Results on Model Financial Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>SF-ML</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>AF-ML</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Customer Controls</td>
</tr>
<tr>
<td>Seller Fixed Effect</td>
</tr>
<tr>
<td>Date Fixed Effect</td>
</tr>
<tr>
<td>Observations</td>
</tr>
</tbody>
</table>

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.

#### 6.1. Financial Performance

We begin by presenting the GMV per customer visit generated by each of the three approaches. We define $Revenue_{PerVisit_{kt}}$ to be the revenue generated from customer $t$’s visit to the coupon sub-page of seller $k$. Panel A of Table 2 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 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:

$$\text{RevenuePerVisit}_{kt} = \alpha_0 + \alpha_1 \text{Approach}_t + X_t + X_k + D_t + \epsilon_{kt}. \quad (3)$$

In the expression above, $\text{Approach}_t$ 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 2 gives the results from specification (3). 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 provide evidence for why this might be the case and also explore where there is potential for improvement with regard to the MNL approach.
6.2. Purchase Probability Accuracy

One explanation for the superior financial performance of the MNL approach is that the fitted MNL choice model may simply be better at predicting purchase probabilities. In order to test whether this in fact is the case, we first must choose a metric to assess the accuracy of the fitted models. One potential metric that could be used 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 well the fitted models are able to predict the product that the arriving customer ultimately purchased.

The first metric is ClassificationAccuracy. For each approach, this metric is the fraction of customers in the hold-out data set where the fitted model’s predicted purchase probability for the product that was purchased (or any of the products that were purchased in the case where multiple products are purchased) is the largest among all displayed options, not including the no-purchase option. The second metric is AverageRank. To compute this metric, we first compute the purchase probabilities of each displayed option (again not including the no-purchase option) under the fitted model. Then, for each customer visit we sort the displayed options in descending order of purchase probability and find the rank of the products that were purchased. We use the convention that the product with the highest predicted purchase probability is assigned rank 1, the second highest is assigned rank 2, and so on. The value we report is the average rank of the purchased options over all customer arrivals in the hold-out data set.

We use 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. 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 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

---

8 We ignore the no-purchase option because the MNL choice model and machine learning algorithms are fit using sales data where only approximately 2% of customers make a purchase. Consequently, for each customer arrival, the fitted models will overwhelmingly predict that the no-purchase option will be selected, leading to a classification accuracy of 0 for observations in which there is a purchase.
Table 3  Model Prediction Performance

<table>
<thead>
<tr>
<th>Panel A: Summary Statistics of Prediction Performance on Purchases</th>
<th>MNL</th>
<th>SF-ML</th>
<th>MNL</th>
<th>AF-ML</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClassificationAccuracy</td>
<td>36.31%</td>
<td>74.55%</td>
<td>36.31%</td>
<td>77.50%</td>
</tr>
<tr>
<td>Difference (All p-values)</td>
<td>38.24% (&lt; 0.0001)</td>
<td>41.19% (&lt; 0.0001)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AverageRank</td>
<td>2.51</td>
<td>1.51</td>
<td>2.51</td>
<td>1.43</td>
</tr>
<tr>
<td>Difference (All p-values)</td>
<td>1.00 (&lt; 0.0001)</td>
<td>1.08 (&lt; 0.0001)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Observations</td>
<td>82,957</td>
<td>68,395</td>
<td>82,957</td>
<td>86,238</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Panel B: OLS Regression Results on Model Prediction Performance</th>
<th>ClassificationAccuracy</th>
<th>AverageRank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Depend</td>
<td>SF-ML</td>
<td>AF-ML</td>
</tr>
<tr>
<td>variable:</td>
<td>(1)</td>
<td>(2)</td>
</tr>
<tr>
<td>SF-ML</td>
<td>0.408****</td>
<td>-1.075****</td>
</tr>
<tr>
<td></td>
<td>(0.003)</td>
<td>(0.007)</td>
</tr>
<tr>
<td>AF-ML</td>
<td>0.437****</td>
<td>-1.152****</td>
</tr>
<tr>
<td></td>
<td>(0.003)</td>
<td>(0.006)</td>
</tr>
<tr>
<td>Buyer Controls</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Seller Fixed Effect</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Date Fixed Effect</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Observations</td>
<td>237,417</td>
<td>237,417</td>
</tr>
</tbody>
</table>

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.

A 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 3. 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 3 gives the accuracy of the fitted model for each of the three approaches based on the two metrics described above. 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 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. AveragePurchaseRank$_{kt}$ is the average rank of purchased products for customer $t$'s visit to seller $k$. We use the following OLS regression specification:

$$\text{TopPurchased}_{kt} = \alpha_0^2 + \alpha_1^2 \text{Approach}_t + X_t + X_k + D_t + \epsilon_{kt}$$ (4)

$$\text{AveragePurchaseRank}_{kt} = \alpha_0^3 + \alpha_1^3 \text{Approach}_t + X_t + X_k + D_t + \epsilon_{kt}$$ (5)

where the set of controls is the same as in specification (3). Our results all hold true if we cluster standard errors at both the customer and seller levels or employ a logistic regression on TopPurchased$_{kt}$ (a binary dependent variable).

Columns (1) and (2) in Panel B of Table 3 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.

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 our accuracy metrics. Consequently, we still 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 4 Mechanism Behind MNL-based Model’s Superior Financial Performance

Panel A: Summary Statistics of Mechanisms

<table>
<thead>
<tr>
<th></th>
<th>MNL</th>
<th>SF-ML</th>
<th>MNL</th>
<th>AF-ML</th>
</tr>
</thead>
<tbody>
<tr>
<td>RevenuePerVisit (RMB)</td>
<td>5.17</td>
<td>4.04</td>
<td>5.17</td>
<td>5.16</td>
</tr>
<tr>
<td>Difference (All p-values)</td>
<td>1.13 (&lt; 0.0001)</td>
<td>0.01 (0.8346)</td>
<td>2.39%</td>
<td>1.96%</td>
</tr>
<tr>
<td>PurchaseIncidence</td>
<td>0.43 (&lt; 0.0001)</td>
<td>-0.1 (&lt; 0.0001)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Observations</td>
<td>3,469,129</td>
<td>3,484,555</td>
<td>3,469,129</td>
<td>3,467,965</td>
</tr>
<tr>
<td>PricePerPurchase</td>
<td>216.2</td>
<td>206.1</td>
<td>216.2</td>
<td>207.3</td>
</tr>
<tr>
<td>Difference (All p-values)</td>
<td>10.1 (&lt; 0.0001)</td>
<td>9.9 (&lt; 0.0001)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Observations</td>
<td>82,957</td>
<td>68,395</td>
<td>82,957</td>
<td>86,238</td>
</tr>
</tbody>
</table>

Panel B: OLS Regression Results on Model Financial Performance

<table>
<thead>
<tr>
<th></th>
<th>Revenue</th>
<th>PurchaseIncidence</th>
<th>PricePerPurchase</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(1)</td>
<td>(2)</td>
<td>(3)</td>
</tr>
<tr>
<td>SF-ML</td>
<td>-0.987***</td>
<td>-0.004***</td>
<td>-6.349***</td>
</tr>
<tr>
<td></td>
<td>(0.073)</td>
<td>(0.0001)</td>
<td>(1.820)</td>
</tr>
<tr>
<td>AF-ML</td>
<td>0.032</td>
<td>0.001***</td>
<td>-5.895***</td>
</tr>
<tr>
<td></td>
<td>(0.077)</td>
<td>(0.0001)</td>
<td>(2.069)</td>
</tr>
<tr>
<td>Buyer Controls</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Seller Fixed Effect</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Date Fixed Effect</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Observations</td>
<td>10,410,587</td>
<td>10,410,587</td>
<td>237,417</td>
</tr>
</tbody>
</table>

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 during our experiment, March 12, 2018 to March 18, 2018. Panel B reports 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 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 PricePerPurchase_{kt} as the average price of the purchased products during customer t’s visit to seller k.

Panel A of Table 4 shows the RevenuePerVisit_{kt}, PurchaseIncidence_{kt}, and 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 4. 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 4 reports the regression results, controlling for customer characteristics, seller fixed effects, and date fixed effects and using specifications similar to specification 3. The regression results generate the same insights as those in Panel A of Table 4.

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 MultiPurchaseCount\(_{k}\) to be the number of visits to seller \(k\) in which the customer makes multiple purchases. Second, we define 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:

\[
RevenuePerVisit_{kt} = \alpha_4 + \alpha_4^\text{Approach}_t + \alpha_4^\text{Moderating Factor}_k + \\
\alpha_4^\text{Approach}_t \times \text{ModeratingFactor}_k + X_t + X_k + D_t + \epsilon_{kt} \quad (6)
\]
Table 5  Heterogeneous Treatment Effect

<table>
<thead>
<tr>
<th></th>
<th>Dependent variable:</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Revenue</td>
<td>(1)</td>
<td>(2)</td>
</tr>
<tr>
<td>SF-ML</td>
<td>-0.822****</td>
<td>-0.780****</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(0.068)</td>
<td>(0.065)</td>
<td></td>
</tr>
<tr>
<td>SF-ML × MultiPurchaseCount</td>
<td>0.030***</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(0.012)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SF-ML × Click-to-Purchase</td>
<td>0.067*</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(0.035)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Customer Controls</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Seller Fixed Effects</td>
<td>No</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Date Fixed Effects</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Observations</td>
<td>5,326,664</td>
<td>5,326,664</td>
<td></td>
</tr>
</tbody>
</table>

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 6.

where ModeratingFactor$_i$ ∈ {Click-to-Purchase$_k$, 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.$^9$

Table 5 reports the results of our heterogeneous treatment analyses. Column (1) of Table 5 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 MultiPurchaseCount$_k$ (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, we implement the first full-scale customer-choice-model-based product recommendation system. We find that our MNL-based approach generates 28% higher revenue per customer

$^9$ Our results are qualitatively robust if we use other numbers of visit cutoffs, such as 500 or 1000.
visit compared to the machine-learning-based 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. As described in Section 4, 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


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


### Appendices

#### A. Proofs

**A.1. Proof of Theorem 1**

*Proof.* To 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

\[
\sum_{j \in N} v_{j\ell}(r_{j} - \hat{z})\hat{y}_{j} = \hat{z}
\]
\[ \sum_{j \in N} v_j r_j \hat{y}_j = \hat{z}(1 + \sum_{j \in N} v_j) \]
\[ \implies R(\hat{y}) = \hat{z}, \]

where the final implication results by dividing both sides of the equality by \(1 + \sum_{j \in N} v_j\). Next we show that \(R(y^*) \leq \hat{z}\). Since \(f(\hat{z}) = \hat{z}\) and \(y^*\) is feasible to problem (2), we have that

\[ \sum_{j \in N} v_j r_j y^*_j \leq \hat{z} \]
\[ \implies \sum_{j \in N} v_j r_j y^*_j \leq \hat{z}(1 + \sum_{j \in N} v_j) \]
\[ \implies R(y^*) \leq \hat{z}. \]

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 \(R(\hat{y}) = R(y^*)\).

A.2. Proof of Proposition 1

Proof. We begin by showing that the while loop runs at most \(L = O(\log(n^2))\) 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 run 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 run 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 \(O(n^2)\).