http://pubsonline.informs.org/journal/opre

OPERATIONS RESEARCH
Vol. 70, No. 1, January–February 2022, pp. 309–328
ISSN 0030-364X (print), ISSN 1526-5463 (online)

Crosscutting Areas

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

Jacob Feldman,[a] Dennis J. Zhang,[a] Xiaofei Liu,[b] Nannan Zhang[b]

[a] Olin Business School, Washington University in St. Louis, St. Louis, Missouri 63130; [b] Alibaba Group Inc., Hangzhou 311100, China

Contact: jbfeldman@wustl.edu, https://orcid.org/0000-0002-5576-1953 (JF); denniszhang@wustl.edu (DJZ); xiaofei.liu@alibaba.com (XL); nannan.zhang@alibaba.com (NZ)

Received: November 5, 2019
Revised: November 18, 2019; August 25, 2020; February 2, 2021
Accepted: April 6, 2021
Published Online in Articles in Advance: October 26, 2021

Area of Review: OR Practice

https://doi.org/10.1287/opre.2021.2158

Abstract. 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. We conducted a large-scale field experiment, in which we randomly assigned 10,421,649 customer visits during a one-week-long period to one of the two approaches and measured the revenue generated per customer visit. The first approach we tested was Alibaba's current practice, which embeds product and customer features within a sophisticated machine-learning algorithm 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. Our second approach, which we developed and implemented in collaboration with Alibaba engineers, uses a featurized multinomial logit (MNL) model to predict purchase probabilities for each arriving customer. We used historical sales data to fit the MNL model, and then, for each arriving customer, we solved a cardinality-constrained assortment-optimization problem under the MNL model to find the optimal set of products to display. Our field experiments revealed that the MNL-based approach generated 5.17 renminbi (RMB) per customer visit, compared with the 4.04 RMB per customer visit generated by the machine-learning-based approach when both approaches were given access to the same set of the 25 most important features. This improvement represents a 28% gain in revenue per customer visit, which corresponds to a 4 million RMB improvement over the week in which the experiments were conducted. Motivated by the results of our initial field experiment, Alibaba then implemented a full-featured version of our MNL-based approach, which now serves the majority of customers in this setting. Using another small-scale field experiment, we estimate that our new MNL-based approach that utilizes the full feature set is able to increase Alibaba's annual revenue by 87.26 million RMB (12.42 million U.S. dollars).

Supplemental Material: The online appendix is available at https://doi.org/10.1287/opre.2021.2158.

Keywords: choice models • product assortment • machine learning • field experiment • retail operations

1. Introduction

Developing a practical approach for most operational problems in retail generally requires building the following two-step schema. First, a retailer or manager must select an underlying demand model whose parameters are estimated from historical sales data. Then, the fitted demand model seeds a subsequent optimization problem whose solution guides key operational decisions, such as inventory levels, prices, or assortments. In the operations and revenue-management communities, the typical approach is to consider demand models that assume a simple and explicit relationship between important product features (such as price) and demand. The simple nature of these demand models not only makes estimation easy, but it also allows for a concrete and nuanced optimization problem to be formulated. From here, the retailer is then tasked with developing tractable and provably good algorithms for the optimization problem of interest.

With the relatively recent rise of machine learning, there is now a clear alternative approach for the initial estimation phase: use machine learning to estimate demand. It is well-documented that algorithms such as gradient-boosted decision trees and neural networks are powerful tools for prediction that could significantly outperform simpler models (like, for example, a choice model) in terms of their ability to capture customer demand patterns. Furthermore, given the wide array of easy-use open-source machine-learning software packages, it is no surprise that many recommendation systems in industry rely on

machine-learning-based solutions (Davidson et al. 2010, Naumov et al. 2019).

Given the discussion above, it is only natural to ask the following question: If it is indeed the case that cutting-edge machine-learning models used in industry can significantly outperform choice models in terms of their ability to accurately predict customer purchasing patterns, then why would a retailer ever adopt the latter? The answer, we find, is that accurate predictions alone are not enough to guarantee that subsequent operational decisions made from these estimates will be profitable. Of equal importance, interestingly, is the sophistication with which the subsequent optimization problem captures key operational trade-offs.

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

The efficacy of the two approaches was assessed by using a large-scale field experiment that spanned one week in March of 2018 and involved more than 10 million customer visits. In this experiment, we compare the performance of our MNL-based recommender system with that of the current machine-learning-based system when both approaches use the same set of the 25 most descriptive product and customer features to parameterize their respective demand models. A full-feature machine-learning-based approach, which utilizes hundreds of features, was also tested and used as a lofty benchmark for the 25-feature MNL-based approach. This full-feature version is exactly the system that was used by Alibaba in practice at the time of our experiments.

1.1. Contributions

We begin by summarizing the insights we derive from the results of our field experiment.

- First and foremost, we observed that the fitted machine-learning models produced far more accurate estimates of the purchase probabilities than the fitted MNL models, yet the MNL-based approach generated 28% higher revenue per visit. This represents a 4 million renminbi (RMB) improvement over just the week in which we conducted our experiments.
- We find that one explanation for the improved financial performance of the MNL-based approach is that customers who experience assortments dictated by the MNL-based approach have an average purchasing price that is 5% higher than that of customers whose recommendations are guided by the machine-learning-based approach. In other words, the MNL-based approach seems to be selecting assortments that lead to purchases of higher-priced products. Furthermore, using heterogeneous treatment effects, we find that the MNL approach's advantages wane for sellers where customers often purchase multiple distinct items. Indeed, when customers generally purchase multiple different items from a single seller, we contend that the offer sets are more likely to consist of complements (or products that are not direct substitutes), and, hence, any advantages gained from modeling substitution pattern are likely diminished.
- Finally, in September 2018, we conducted another small-scale field experiment consisting of 2 million customer visits, in which full-feature versions of both approaches were implemented and compared. We found that our MNL-based approach generated 3.37% higher revenue per visit compared with Alibaba's current machine-learning-based recommender system. Using data from both experiments, we estimate that using the full-feature MNL-based approach would improve Alibaba's annual revenue by 87.26 million RMB (12.42 million U.S. dollars (USD)). Unfortunately, unlike the initial experiment, we were not given access to individual visit-level data for this second experiment due to data-security issues. As a result, we were neither able to measure the accuracy of each fitted model nor able to perform any analysis using heterogeneous treatment effects. It is for this reason that we focus on the initial set of experiments in the main body of the paper and relegate the results of this follow-up experiment to Online Appendix 11.

In addition to the findings unveiled by our field experiments, we also make the following algorithmic contributions:

- An important component of our MNL-based approach is a novel implementation of the algorithm proposed in Rusmevichientong et al. (2010) for solving cardinality-constrained assortment problems under the MNL model. Specifically, our updated version of this

algorithm improves the running time by a factor of $\log(n)$, where n denotes the number of products.

- Our second contribution comes in the form of a novel approximation scheme for a special constrained version of the assortment problem under the MNL model. Our field experiments only consider the problem of choosing optimal six-product displays; however, there are other operational levers that Alibaba could exploit in this discount-coupon setting to increase revenues. In Section 4.3, we consider two such levers, namely, price and icon size. For the pricing problem, Alibaba must simultaneously decide which products to offer as well as the prices to charge for each of these offered products. For the icon-size problem, we allow for Alibaba to choose the size of the icon representing each displayed product and enforce a limit on the total available screen space.

1.2. Organization

The majority of the remaining content is centered around providing all of the necessary details of our field experiments. Along these lines, in Section 2, we provide a detailed overview of the discount-coupon setting that we consider on Alibaba, which is the canvas for our field experiments. Sections 3 and 4 describe how we address the respective estimation and assortment problems within the two approaches that we test. The full details and results of our first experiment are given in Sections 5 and 6, respectively.

1.3. Related Literature

There is an expansive collection of previous works regarding customer choice models and their accompanying assortment and estimation problems. As such, we decide to only review past work that focuses on the MNL model, because this model is the focus of our work. We also include a summary of past work on recommender systems and retailing operations.

1.3.1. The MNL Choice Model. As mentioned earlier, the MNL model was originally was conceived by Luce (1959), and its practical use later was most notably established by McFadden (1974), who, among other results, showed that the log-likelihood function is concave in the model parameters. To the best of our knowledge, Vulcano et al. (2012) was 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 developed an iterative expectation-maximization (EM) approach that is based on uncensoring the most preferred product of each customer. Later, Vulcano and Abdullah (2020) used a similar minorization-maximization (MM) algorithm to estimate the parameters of an MNL model from historical transaction data. They showed that this newly proposed technique produces accurate estimates while being computationally superior to the previously mentioned EM approach.

With regards to works concerning the assortment problem under MNL preferences, the seminal papers of Talluri and van Ryzin (2004) and Gallego et al. (2004) established that the optimal assortment must consist of some subset of the highest-revenue products in the unconstrained setting. When a cardinality constraint is added to the assortment problem, Rusmevichientong et al. (2010) provided a purely combinatorial polynomial-time algorithm, which is able to identify the optimal assortment. Building on this result, Sumida et al. (2019) showed 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.

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

1.3.3. Retailing Operations. Lastly, our paper relates closely to the literature that studies assortment (Caro and Gallien 2007, Li and Talluri 2019, Gallego et al. 2020), inventory (Netessine and Rudi 2003, Chen and Simchi-Levi 2004, Rumyantsev and Netessine 2007, Caro and Gallien 2010, Chen et al. 2014, Ramachandran et al. 2018), and pricing (Elmachtoub and Hamilton 2017, Tereyağoğlu et al. 2017, Cui et al. 2018, Cui et al. 2019) 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. (2019)

estimate the impact of inventory on sales at car dealerships and propose an inventory policy to maximize variety. Golrezaei et al. (2014) consider a dynamic assortment problem in which a retailer is allowed to personalize the assortment of products offered to each arriving customer in response to just-revealed features and the current inventory levels of each product.

2. Alibaba's Retail Setting and Product-Display Problem

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. More specifically, we focus on a product-recommendation setting 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 subpage that contains six displayed products, each of which can be purchased at a discount using the coupon. Alibaba chooses to display only six products because this is the largest number of products that can be displayed within a single page on a mobile device. Figure 1 shows how a customer progresses from a seller's front page to the coupon subpage to the six displayed products. As of March 2018 (just before our experiment), there were approximately 250,000 sellers on the Alibaba platform who offered the mobile coupon discounts. On a weekly basis, these sellers witness more than 25 million unique page views on their coupon subpages and generate more than RMB 127 million (equivalent to USD 20 million) in GMV.

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 subpage. Sellers on the Alibaba platform typically have between 100 and 2,000 unique SKUs, all of which are in the same product category and, hence, can be loosely considered substitutes (we investigate this claim later in Section 6.4). 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.

3. The Estimation Problem

In this section, we describe the two approaches used to estimate the purchase probabilities P_{jt} that seed the *Alibaba Product Display Problem*. The first approach embeds product and customer features within sophisticated machine-learning algorithms. This approach is Alibaba's current practice for solving the estimation problem. The second approach fits featurized MNL models to the historical sales data using maximum-likelihood estimation (MLE). Although the latter approach can be described in full detail, we are not able to provide the exact details of the machine-learning-based approach due to confidentiality concerns; however, we intend to provide enough details so that the advantages and drawbacks of this approach can be well understood. Furthermore, in Online Appendix 8, we provide a case study comparing the fitting accuracy of an off-the-shelf machine-learning algorithm with that of the MNL model using historical sales data from the top 10 sellers (based on traffic) from Tmall.com. The intent of this case study is to formally define various accuracy metrics that we utilize to assess our fitted models and also to show that it is not difficult to develop machine-learning-based estimation schemes that outperform the MNL model in terms of these metrics.

3.1. Available Sales Data and Product/Customer Features

Before diving into the details of either approach, we first discuss the makeup of the available historical sales data used to fit the machine-learning and MNL models. This training data are composed of historical sales information from τ past customers, each of whom is shown six products. For each arriving customer t, we let $S_t \subseteq \mathcal{N}$ be the six displayed products, which the system stores as vectors of representative feature values. The product features that are used include high-dimensional static features, such as a one-hot encoding representation of product ID and seller ID, in addition to low-dimensional static features, such as product category. Dynamic product features, such as the number and extent of customer reviews, are also included in the feature set. These features are updated constantly based on customer interactions.

Figure 1. (Color online) The Process of Landing on Our Recommendation Page

Finally, we note that product features are also engineered from product descriptions and pictures. For example, there is a feature associated with the image quality of each product's icon that is displayed to customers within the app. The system also records an associated feature vector that describes the characteristics of the customer at hand. The customer-specific features include demographic information, such as age, gender, and registration time. Other customer features are descriptive of past behaviors within the app—for example, the number of products viewed, collected, purchased, and returned in the past.[1]

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 a particular product will appeal to a 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. Because these collaborative filtering scores depend on customer behavior within the app, they are dynamically updated so that they reflect current trends. In total, hundreds of features—numerical and categorical, static and dynamic—are available to be used within the estimation schemes. As noted above, we will mainly focus on implementations of both approaches that use only the top 25 features. In Section 6.2, we provide evidence for the notion that these 25 features are, in fact, quite descriptive by showing that the fitting accuracy of the machine-learning models improves only slightly as we move from the 25-feature to the full-feature models.

3.2. Fitting Machine-Learning Models

In what follows, we formalize the machine-learning-based approach for estimating the purchase probabilities P_{jt}. Each observation within the training data set can be described as a triple $(\mathbf{X_{jt}}, c_{jt}, z_{jt})$ corresponding to a specific arriving customer t and displayed product $j \in S_t$. The vector $\mathbf{X_{jt}}$ gives the features associated with the particular observation, while the response variables $c_{jt}, z_{jt} \in \{0,1\}$, respectively, indicate whether customer t clicked and purchased displayed product j. In total, the training data (or purchase history) consist of $T = 6\tau$ observations, which we represent as $\mathcal{PH}_{ML} = \{(\mathbf{X_{jt}}, c_{jt}, z_{jt}) : t = 1, \ldots, T, j \in S_t\}$. We note that, for this approach, each observation $(\mathbf{X_{jt}}, c_{jt}, z_{jt}) \in \mathcal{PH}_{ML}$ does not encode the set of products that were offered alongside product j to customer t. As such, the estimates of the purchase probabilities are independent of the assortment of products displayed and, hence, do not account for customer substitution behaviors. We note that it is indeed possible to build a machine-learning model in which the estimated demand for a particular product is a function of the availability of other products (i.e., incorporating substitution or complementarity effects); however, this fitted model will not provide

an explicit relationship between assortments and purchasing probabilities that is needed to formulate a tractable assortment problem. As a result, the fitted machine-learning models must assume that the demand for each product is independent of the other offered products.

The training data are used to solve two independent estimation problems, which are then combined to form estimates of the purchase probabilities P_{jt}. First, the training data are used to derive estimates of the click probabilities $\mathbb{P}(c_{jt} = 1)$, which represent the probability that customer t will click on product j. To do so, various machine-learning algorithms are employed, which are finely tuned to match the past click history described in $\mathcal{PH}_{ML}$. We let the output of this estimation procedure be a function $f(\mathbf{X_{jt}})$, 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 $g(\mathbf{X_{jt}})$, which produces accurate estimates of the conditional purchase probabilities $\mathbb{P}(z_{jt} = 1 | c_{jt} = 1)$. Ultimately, Alibaba uses $P_{jt}(\mathbf{X_{jt}}) = f(\mathbf{X_{jt}}) \cdot g(\mathbf{X_{jt}})$ as their estimates of the purchase probabilities, where we now explicitly express this probability as a function of the feature vector $\mathbf{X_{jt}}$.

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

3.3. Fitting The MNL Model

In this section, we formally introduce the MNL choice model and describe how its underlying parameters are fit to historical sales data. We refer the interested reader to Train (2009) for a formal derivation of the MNL model from first principles.

3.3.1. The MNL Choice Model.

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 a Gumbel random variable denoted as E_{jt}, which are assumed to be independent and identically distributed for each product. More formally, for each product $j \in S_t$, we have that $U_{jt} = V_{jt} + \epsilon_{jt}$. In order to incorporate product and customer features within the above utility function, one can write the deterministic component of the utility as $V_{jt} = \boldsymbol{\beta}^T \mathbf{X_{jt}}$, where the vector $\mathbf{X_{jt}}$ denotes the values of the relevant features for customer t and product j, and $\boldsymbol{\beta}$ is a vector of feature weights. If the retailer offers assortment $S_t \subseteq N$ to customer t, then the probability that product $j \in S_t$ is purchased is given by

$$P_{jt}(S_t, X_t) = \mathbb{P}\left[U_{jt} = \max_{i \in S_t} U_{it}\right] = \frac{e^{\boldsymbol{\beta}^T \mathbf{X_{jt}}}}{1 + \sum_{i \in S_t} e^{\boldsymbol{\beta}^T \mathbf{X_{jt}}}},$$

where $X_t = \{\mathbf{X_{jt}} : j \in S_t\}$ 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.

3.3.2. Fitting the MNL Choice Model.

We use maximum-likelihood estimation to derive estimates for the $\boldsymbol{\beta}$ coefficients. We formulate the log-likelihood using historical sales data from τ customers. More specifically, we represent the past purchasing history of the τ customers as the set $\mathcal{PH}_{MNL} = \{(S_t, X_t, z_t) : t = 1, \ldots, \tau\}$, where we note again that S_t denotes the set of six displayed products and $X_t = \{\mathbf{X_{jt}} : j \in S_t\}$ 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.

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

$$\max_{\boldsymbol{\beta}} \mathcal{LL}(\boldsymbol{\beta};\ \mathcal{PH}_{MNL}), \tag{1}$$

where

$$\mathcal{LL}(\boldsymbol{\beta};\ \mathcal{PH}_{MNL}) = \sum_{t=1}^{\tau} \left(\boldsymbol{\beta}^T X_{z_t t} - \log\left(1 + \sum_{j \in S_t} e^{\boldsymbol{\beta}^T \mathbf{X_{jt}}}\right)\right).$$

It is well known that the objective function in (1) is concave in $\boldsymbol{\beta}$ (McFadden 1974). Exploiting this fact, 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) because $\tau \approx 20 - 30$ million.

4. The Assortment Problem

In this section, we consider the assortment problem that results when the fitted MNL and machine-learning models are used to seed the *Alibaba Product Display Problem*. Furthermore, in Section 4.3, we also consider natural practical extensions of the cardinality-constrained assortment problem under the MNL model and develop a novel approximation scheme for one of these extensions.

4.1. The Machine-Learning Fits

After fitting the machine-learning models, we are able to derive estimates $P_{jt}(\mathbf{X_{jt}})$ of the purchase probabilities for any customer t and product j. In this case, 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 $r_j \cdot P_{jt}(\mathbf{X_{jt}})$ and then selects the top six products in this ordering. This algorithm is trivially optimal because the purchase probabilities do not depend on the set of offered products. This simple greedy approach easily runs within the 200-millisecond threshold enacted to avoid page delays.

4.2. The MNL Fits

Next, we consider the cardinality-constrained assortment problem that results when the purchase probabilities P_{jt} in the *Alibaba Product Display Problem* are dictated by our fitted MNL choice model. Again, we consider a setting with n products indexed by the set $\mathcal{N} = \{1, \ldots, n\}$, 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 $v_{jt} = e^{\boldsymbol{\beta}^{*T}\mathbf{x}_{jt}}$, where $\boldsymbol{\beta}^*$ is the optimal solution to Problem (1). We will encode our assortment decision through the binary vector $\mathbf{y} \in \{0,1\}^n$, where we set $\mathbf{y}_j = 1$ if product j is offered and $\boldsymbol{y}_j = 0$ otherwise. With this notation in hand, the purchase probability of product j under assortment $\mathbf{y}$ is given by $P_j(\mathbf{y}) = v_j\mathbf{y}_j/1 + \Sigma_{i\in\mathcal{N}} v_i\mathbf{y}_i$, where we drop the preference weight's dependence on t (and X_t) for ease of notation in the remainder of this section. The expected revenue of assortment $\mathbf{y}$ can therefore be expressed as

$$R(\mathbf{y}) = \sum_{j\in N} P_j(\mathbf{y}) \cdot r_j = \frac{\sum_{j\in\mathcal{N}} r_j v_j \mathbf{y}_j}{1 + \sum_{i\in\mathcal{N}} v_i \mathbf{y}_i}.$$

Finally, we denote the set of feasible assortments as $\mathcal{F} = \{\mathbf{y} \in \{0,1\}^n : \Sigma_{j=1}^n \mathbf{y}_j = 6\}$. The cardinality-constrained assortment problem of interest is therefore:

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

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

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

$$f(z) = \max_{\mathbf{y}\in\mathcal{F}} \sum_{j\in N} v_j(r_j - z)\mathbf{y}_j, \qquad (2)$$

and we

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

For any 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 $j \in N$, and note that $\hat{y}(z)$ will trivially consist of the six products with the largest contributions. The following theorem, which has appeared in one form or another in numerous assortment optimization papers (Rusmevichientong et al. 2010, Davis et al. 2014, Sumida et al. 2019), elucidates the importance of (2). For completeness, we include a proof of this result in the online appendix, which hosts all other proofs as well.

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

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 $\hat{y}(\hat{z})$. Hence, all that remains is to describe an efficient process for finding $\hat{z}$. Rusmevichientong et al. (2010) cleverly note that because the assortments $\hat{y}(z)$ only depend on the relative orderings of the contributions of each product, then 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 $i, j \in N$. Furthermore, because the contribution of each product is a linear function of z, there can be at most $O(n^2)$ intersection points because each of the n lines can intersect one of the other $n-1$ lines at most once.

Algorithm 1 (Solving *MNL-Card*)

1. $z_- \leftarrow 0$
2. $z_+ \leftarrow \max_{i\in N} r_i$
3. $\mathcal{Z} = \{z_-, z_+\}$
4. **for** $i \in \mathcal{N}$ **do**
5. **for** $j \in \mathcal{N} : j \neq i$ **do**
6. $z(i,j) = \frac{v_i r_i - v_j r_j}{v_i - v_j}$
7. $\mathcal{Z} = \mathcal{Z} \cup z(i,j)$
8. **end for**
9. **end for**
10. $t = 0$
11. $\bar{Z}_t \leftarrow \mathcal{Z}$
12. **while** $|\bar{Z}_t| > 2$ **do**

13. $z_m \leftarrow Med(\bar{Z}_t)$
14. $f \leftarrow f(z_m)$
15. **if** $f < z_m$ **then**
16. $z_- \leftarrow z_m$
17. $\bar{Z}_{t+1} \leftarrow \{z \in \bar{Z}_t : z \le z_m\}$.
18. **else**
19. $z_+ \leftarrow z_m$
20. $\bar{Z}_{t+1} \leftarrow \{z \in \bar{Z}_t : z > z_m\}$
21. **end if**
22. $t \leftarrow t+1$
23. **end while**
24. $z_{\text{final}} \leftarrow Med(\bar{Z}_t)$
25. **return** $\hat{y}(z_{\text{final}})$

More formally, for products $i,j \in N$, we let $z(i,j) = (v_i r_i - v_j r_j)/(v_i - v_j)$ be the value of z satisfying $c_i(z) = c_j(z)$. We denote the set of all such intersection points as $\mathcal{Z} = \{z(i,j) : i,j \in N\} \cup \{0\}$ and note that this set can be constructed in $O(n^2)$ by simply enumerating all pairs of products. The candidate assortments can then be captured through the set $\mathcal{Y} = \{\hat{y}(z) : z \in \mathcal{Z}\}$, and, based on the discussion above, we know that $\mathbf{y}^* = \arg\max_{\mathbf{y}\in\mathcal{Y}} R(\mathbf{y})$. From a computational perspective, the most burdensome step is that of computing the set of candidate assortments $\mathcal{Y}$. To see this, note that for each $z \in \mathcal{Z}$, in order to compute the assortment $\hat{y}(z)$, we must compute the relative ordering of the contributions $c_i(z)$ for each product $i \in N$. Rusmevichientong et al. (2010) show that by first sorting $\mathcal{Z}$, these relative orderings can be computed in a running time of $O(n^2)$. Hence, the bottleneck step is the initial sorting operation of the $O(n^2)$ intersection points of $\mathcal{Z}$, which requires $O(n^2\log(n))$ operations. In what follows, we give an algorithm that finds $\mathbf{y}^*$ in a running time of $O(n^2)$. We do so 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 $\hat{z}$. This approach is fully spelled out in Algorithm 1, and its running time is formally established in the following proposition.

Proposition 1. *The running 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 easily runs within the 200-millisecond threshold and has never timed out. Furthermore, in Online Appendix 9, we conduct numerical experiments using randomly generated instances of *MNL-Card*, which show that our updates to the original algorithm have the potential to improve running times by a factor of two or three.

4.3. Practical Extensions of the MNL Assortment Problem

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

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

$$\mathcal{F}_1 = \{\mathbf{y} \in \{0,1\}^{m\times n} : \sum_{j=1}^{m} \mathbf{y}_{ij} \le 1 \ \ \forall i \in N\}$$

$$\cap \{\mathbf{y} \in \{0,1\}^{m\times n} : \sum_{i=1}^{n}\sum_{j=1}^{m} \mathbf{y}_{ij} = 6\},$$

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

$$\max_{\mathbf{y}\in\mathcal{F}_1} R(\mathbf{y}) = \max_{\mathbf{y}\in\mathcal{F}_1} \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} r_i v_{ij}\mathbf{y}_{ij}}{1+\sum_{i=1}^{n}\sum_{j=1}^{m} v_{ij}\mathbf{y}_{ij}}. \tag{3}$$

Exploiting the fact that $\mathcal{F}_1$ can be encoded as totally unimodular constraint matrix, Sumida et al. (2019) show that Problem (3) can be formulated as a tractable linear program. We recount this formulation in the online appendix for the sake of completeness and, more importantly, because it will be an essential component of the novel algorithm we develop to tackle the icon-display-size extension that is presented next.

2. *Icon display size*: Here, we consider the problem of optimally choosing the size of the icons corresponding to each displayed product. Note that in the current set-up, the icons of all six displayed products are the same size (see Figure 1), but in other settings on Alibaba, the

icon sizes can differ in size, which motivates us to consider this extension. To model our updated setting in which product icon sizes do not have to be homogeneous, we assume that the icon of each displayed product can take on one of m sizes indexed by the set $M = \{1,\dots,m\}$ and use c_j to be the screen space consumed by an icon of size j. We assume the total available screen space is C. Furthermore, we use v_{ij} to be the MNL preference weight of product i when it is offered with an icon of size j. In this way, we assume that the preference weight of each product is influenced by the size of the icon used to display this product to customers. We let $\mathbf{y}_{ij} \in \{0,1\}$ be a binary indicator of whether product $i \in N$ is displayed with icon size $j \in M$. In this setting, the feasible assortments can be captured through the set

$$\mathcal{F}_2 = \{\mathbf{y} \in \{0,1\}^{m\times n} : \sum_{j=1}^{m} \mathbf{y}_{ij} \le 1 \ \ \forall i \in N\}$$

$$\cap \{\mathbf{y} \in \{0,1\}^{m\times n} : \sum_{i=1}^{n}\sum_{j=1}^{m} c_j \mathbf{y}_{ij} \le C\},$$

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

$$\max_{\mathbf{y}\in\mathcal{F}_2} R(\mathbf{y}) = \max_{\mathbf{y}\in\mathcal{F}_2} \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} r_i v_{ij}\mathbf{y}_{ij}}{1+\sum_{i=1}^{n}\sum_{j=1}^{m} v_{ij}\mathbf{y}_{ij}}. \tag{4}$$

Unfortunately, $\mathcal{F}_2$ cannot be encoded as a totally unimodularity constraint matrix due to the knapsack constraint that limits the total screen-space consumption. As such, the linear-programming approach of Sumida et al. (2019) is seemingly rendered ineffective. Surprisingly, however, we are to salvage the optimal solution to an "incorrect" linear programming formulation of Problem (4) inspired by Sumida et al. (2019) to produce a feasible assortment that garners an expected revenue of at least one-third of the optimal expected revenue. This notion is formalized in the following theorem.

Theorem 2. *Let* $\mathbf{y}^*_{\text{icon}}$ *be the optimal solution to Problem* (4). *There is a polynomial time algorithm that produces an assortment* $\mathbf{y} \in \mathcal{F}_2$ *that satisfies* $R(\mathbf{y}) \ge \frac{1}{3}\cdot R(\mathbf{y}^*_{\text{icon}})$.

5. Experiment Design and Data

In this section, we first discuss the design of our field experiment. Then, we provide a randomization check to demonstrate the rigor of our experimental design, which includes summary statistics of the raw results data.

5.1. Experiment Design

Our experiment officially started on March 12, 2018, and lasted for two weeks. However, because of security reasons, we can only report the results from the first week.[2] Throughout the experiment, we tested the following three approaches:

1. **The MNL-based approach (MNL approach):** Customers assigned to this approach see six-product displays from the MNL-based approach. This approach parameterizes the underlying MNL model using the 25 most important features.

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 using the same set of the 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 that uses all available features within the machine-learning estimation algorithms. This was the current product-recommendation system used by Alibaba before our experiment.

During the experimental week beginning on March 12, 2018, each customer who arrived at the coupon subpages for any participating seller was randomly assigned one of the three approaches based on a unique hash number derived from the customer's ID and an experiment ID.[3] Each customer was only assigned to one of the three product-recommendation approaches described above, regardless of how many times she visited the coupon subpage.

5.2. Summary Statistics and Randomization Check

Over the week of our experiment, 5 million customers were selected to participate in our experiments, and they collectively generated approximately 10.4 million arrivals to the product-recommendation pages. Among these 5 million customers, 1,879,903 customers were assigned to the MNL approach, 1,879,598 customers were assigned to the SF-ML approach, and 1,876,940 customers were assigned to the AF-ML approach.

Panel A of Table 1 presents customer and seller information from the three experiment groups to confirm that the customers and sellers assigned to each of the three approaches were comparable in terms of demographics, spending habits, and revenue before the experiment. We observe 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.[4] The results of our randomization checks suggest that any differences between the purchasing behaviors of customers within our experiment should be attributed to the specific approach they have been assigned. Panel B of Table 1 shows the aggregate number of page views, clicks,

Table 1. Summary Statistics

	MNL	SF-ML	AF-ML	Min pairwise p-value
Panel A: Randomization check				
Seller Daily GMV in Last 30 Days	1.7 million	1.7 million	1.7 million	>0.3
Seller Number of Products in Last 30 Days	2,187	2,186	2,189	>0.2
Seller Registration Year	2013	2013	2013	>0.4
Customer Registration Year	2012	2012	2012	>0.3
Customer Gender (Male = 1)	0.26	0.26	0.26	>0.5
Customer Age	30.2	30.2	30.3	>0.3
Panel B: Summary statistics				
Number of Page Views	3,469,129	3,484,555	3,467,965	
Number of Products Clicked	421,896	368,987	423,046	
Number of Products Purchased	86,585	70,699	90,033	
GMV (RMB)	18 million	14 million	17.8 million	

Notes. Panel A reports the average monthly GMV, average number of products available to the seller, average seller registration year, 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 approaches). t-tests between the differences in averages of the three approaches have p-values > 0.05 for all pairwise comparisons. Panel B reports the total number of page views, number of products clicked/purchased, and total GMV in each approach.

purchases, and revenue generated by customers assigned to each of the three approaches during the one-week-long experiment.

6. Main Results

In this section, we present the results of our field experiment. We begin by detailing the financial performance of three approaches. The metric Alibaba uses internally to assess a product-recommendation systems is GMV (used synonymously with revenue throughout) per customer visit, and, hence, we also adopt this metric to judge 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.

6.1. Financial Performance

We begin by presenting the GMV per customer visit generated by each of the three approaches. We define $\text{RevenuePerVisit}_{ijk}$ to be the revenue generated from customer i's visit to the coupon subpage of seller j at time k. Table 2 shows the revenue per visit of the MNL, SF-ML, and AF-ML approaches during our experimental period. The first row of Table 2 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). Moreover, the MNL approach's financial performance surprisingly is also on par with that of the AF-ML approach, which uses hundreds of features within its estimation scheme, compared with only 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).

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

$$\text{RevenuePerVisit}_{ijk} = \alpha_0^1 + \alpha_1^1 \text{Approach}_i + \mathbf{X_i} + \mathbf{X_j} + D_k + \epsilon_{ijk}, \tag{5}$$

where $\text{RevenuePerVisit}_{ijk}$ is the revenue generated by customer i's visit to seller j's subpage at time k; Approach_i is a categorical variable indicating the approach to which customer i has been assigned; $\mathbf{X_i}$ and $\mathbf{X_j}$ are 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; D_k is a date-specific fixed effect; and ϵ_{ijk} is the idiosyncratic shock associated with each observation. Because one consumer's visits may be correlated due to budget constraints or spending habits, 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.

Table 3 gives the results from Specification (5). In this specification, we use data from the MNL approach as the baseline, so the coefficients of the SF-ML and

Table 2. Model Financial Performance: Summary Statistics of Financial Performance

	MNL	SF-ML	MNL	AF-ML
RevenuePerVisit (RMB)	5.17	4.04	5.17	5.16
Difference (all *p*-values)	1.13 (<0.0001)		0.01 (0.8346)	
Relative Improvement	28.0%		0.2%	
Observations	3,469,129	3,484,555	3,469,129	3,467,965

Notes. Standard errors are robust and clustered at the customer level. Reported is the average financial performance, in terms of revenue per customer visit, across different algorithms during our experimental period (March 12, 2018–March 18, 2018).

AF-ML approach indicators represent the financial performance difference between the MNL approach and each of the other two approaches. Column (1) of Table 3 does not control for any additional variables, and we successfully recover the mean difference from Table 2: A customer visit under the MNL approach generates 1.126 and 0.015 more RMB per visit compared with 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 Table 3 controls for the customer characteristics, seller fixed effects, and date fixed effects, and our estimates are qualitatively similar with smaller standard errors.

The results described above indicate that the MNL approach performs quite well in relation to both of the machine-learning-based approaches. In what follows, we show that this superior performance cannot likely be explained by superior prediction accuracy, as Alibaba's machine-learning models are far more accurate than the fitted MNL models.

6.2. Purchase-Probability Accuracy

In this section, we compare the fitting accuracies of each fitted model using the experimental sales data that were generated from March 12, 2018, to March 18, 2018, to compute the classification accuracy and average rank metrics that are formally defined in Online Appendix 8. In short, the classification accuracy is a measure of how frequently the fitted model correctly predicts the item that was purchased. The average rank, on the other hand, provides a measure of how far off the predicted choice was from the true choice. This metric ranges from one to n, with a lower rank signifying a more accurate model. Both accuracy metrics are computed by using only data points corresponding to purchases of a single product. In Online Appendix 8, we explain why we ignore no-purchase and multipurchase events in computing the accuracy metrics and also detail why it is not possible to use likelihood measures to assess fitting accuracy.

Table 4 gives the accuracy of the fitted model for each of the three approaches based on the two metrics we consider. The top two rows of Table 4 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 Table 4 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 pairwise differences between these average ranks are statistically significantly (all p-values < 0.0001).

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

$$\text{TopPurchased}_{kt} = \alpha_0^2 + \alpha_1^2 \text{Approach}_t + X_t + X_k + D_t + \epsilon_{kt}, \tag{6}$$

$$\text{AveragePurchaseRank}_{kt} = \alpha_0^3 + \alpha_1^3 \text{Approach}_t + X_t + X_k + D_t + \epsilon_{kt}, \tag{7}$$

Table 3. Model Financial Performance: OLS Regression Results on Model Financial Performance

	Dependent variable	
	Revenue (1)	Revenue (2)
SF-ML	−1.126****	−0.987****
	(0.094)	(0.073)
AF-ML	−0.015	0.032
	(0.110)	(0.077)
Customer Controls	No	Yes
Seller Fixed Effect	No	Yes
Date Fixed Effect	No	Yes
Observations	10,421,649	10,421,649

Notes. Standard errors are robust and clustered at the customer level. Reported are the results from OLS regression that estimate the difference between different models' revenue per customer visit. Column (1) of does not control for any additional control variables, while column (2) controls for customer characteristics, seller fixed effects, and date fixed effects.

****$p < 0001$.

Table 4. Model Prediction Performance: Summary Statistics of Prediction Performance on Purchases

	MNL	SF-ML	MNL	AF-ML
ClassificationAccuracy	36.31%	74.55%	36.31%	77.50%
Difference (all *p*-values)	38.24% (< 0.0001)		41.19% (< 0.0001)	
AverageRank	2.51	1.51	2.51	1.43
Difference (all *p*-values)	1.00 (< 0.0001)		1.08 (< 0.0001)	
Observations	82,957	68,395	82,957	86,238

Notes. Standard errors are robust and clustered at the customer level. Reported is the average prediction power of customers' purchasing behaviors during our experiment.

where the set of controls is the same as in Specification (5). Our results all hold true if we cluster standard errors at both the customer and seller levels or employ a logistic regression on TopPurchased_{it} (a binary dependent variable).

Columns (1) and (2) in Table 5 present results from Specifications (4) and (5). In these specifications, we use the accuracy performance under the MNL approach as a baseline, so the coefficients of the SF-ML and AF-ML indicators represent the difference between the MNL approach and each of the machine-learning approaches. The coefficients of column (1) are all positively significant, showing that both machine-learning approaches have higher prediction accuracy compared with 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 Table 4 (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 have demonstrated that although the MNL approach performs much better than the SF-ML approach and on par with the AF-ML approach in terms of revenue per visit, it actually has significantly worse prediction accuracy than both machine-learning approaches with respect to both accuracy metrics. In the next section, we ascribe the superior financial performance of the MNL approach to its ability to produce six-product displays that ultimately lead to higher-revenue products being purchased.

6.3. Average Purchase Price

In this section, we show that, on average, the MNL-based approach chooses six-product displays that lead to purchases of higher-revenue products. Furthermore, we also find that customers assigned the MNL and AF-ML approach are far more likely to make a purchase than customers assigned the SF-ML approach. To formalize this analysis, we first define $\text{PurchaseIncidence}_{kt}$ as a binary indicator equal to one if customer t's visit to seller k results in a purchase and zero otherwise. We also define $\text{PricePerPurchase}_{kt}$ as the average price of the purchased products during customer t's visit to seller k.

Table 6 shows the $\text{RevenuePerVisit}_{kt}$, $\text{Purchase Incidence}_{kt}$, and $\text{PricePerPurchase}_{kt}$ for all three approaches during our experimental period. The left side of Table 6 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. Furthermore, under the MNL approach, customers, on average, make a purchase 22% more frequently than under the SF-ML approach. Hence, although the SF-ML approach produces more accurate estimates of the purchase probabilities, it is not able to offer assortments that are as desirable or 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 Table 6. We see that the MNL approach leads to a significantly higher average purchasing price (i.e., RMB 216.2 versus RMB 207.3; $p < 0.00001$) and significantly lower purchasing incidence (i.e., 2.39% versus 2.49%; $p < 0.00001$), which ultimately leads to similar revenue performance, as the two metrics balance each other. As a slight side note, Table 6 also reveals that the AF-ML approach outperforms the SF-ML approach because it recommends displays that lead to a higher purchase incidence rate. Table 7 reports the regression results, controlling for customer characteristics, seller fixed effects, and date fixed effects and using specifications similar to Specification (5). The regression results generate the same insights as those in Table 6.[5]

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

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 distinct product, whereas in practice, customers may often purchase multiple unique products. Second, the MNL choice

Table 5. Model Prediction Performance: OLS Regression Results on Model Prediction Performance

	Dependent variable	
	ClassificationAccuracy (1)	AverageRank (2)
SF-ML	0.408****	−1.075****
	(0.003)	(0.007)
AF-ML	0.437****	−1.152****
	(0.003)	(0.006)
Buyer controls	Yes	Yes
Seller fixed effect	Yes	Yes
Date fixed effect	Yes	Yes
Observations	237,417	237,417

Notes. Standard errors are robust and clustered at the customer level. Columns (1) and (2) report the reports from OLS regression on models' prediction power of customers' purchasing behaviors.
****$p < 0001$.

model in its standard form cannot incorporate customer click behavior within how it models customer preferences. Based on these theoretical limitations, we identify two seller characteristics that may influence the performance of the MNL approach. We first define $\text{MultiPurchaseCount}_k$ to be the number of visits to seller k in which the customer purchases multiple distinct products. Second, we define $\text{Click}-\text{to}-\text{Purchase}_k$ as the ratio of the number of clicked products to the number of purchased products across all visits to seller k.

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

$$\begin{aligned}\text{RevenuePerVisit}_{kt} &= \alpha_0^4 + \alpha_1^4 \text{Approach}_t \\ &+ \alpha_2^4 \text{Moderating Factor}_k + \alpha_3^4 \text{Approach}_t \\ &\times \text{ModeratingFactor}_k + X_t + X_k + D_t + \epsilon_{kt}, \end{aligned} \quad (8)$$

where $\text{ModeratingFactor}_i \in \{\text{Click}-\text{to}-\text{Purchase}_k, \text{MultiPurchaseCount}_k\}$. We only focus on the observations corresponding to customers who were assigned to the MNL and SF-ML approaches and from sellers who had at least 100 visits, which comprises 76% of the sellers.[6]

Table 8 reports the results of our heterogeneous treatment analyses. Column (1) of Table 8 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 multipurchase incidences. In other words, our MNL approach performs worse for sellers whose customers are more likely to purchase multiple distinct items from an offer set. Clearly, the MNL-based approach should suffer for these sellers because the MNL model was built to capture single-purchase events only. Moreover, there is also a slightly more nuanced interpretation of these results related to the perceived substitutability of the products available to each seller. More specifically, we contend that the products available to sellers who generally witness fewer multipurchase events are likely close substitutes. Note that if all products were perfect substitutes, then each customer would never need to purchase more than a single product. On the other hand, sellers who see a large number of multipurchase events likely manage a collection of products that are complementary (or at least not close substitutes), which understandably makes the MNL-based approach less effective.

Moving to our other moderating factor, column (2) also reveals a positive interaction term between the SF-ML indicator and Click – to – Purchase. Similarly, this demonstrates that the MNL-based approach performs worse when the ratio of clicks to purchases is high. Columns (1) and (2) collectively show that the theoretical limitations we identified with the MNL approach indeed affect its performance: A one-unit increase in $\text{MultiPurchaseCount}_k$ ($\text{Click}-\text{to}-\text{purchase}_k$) would translate to a 0.03 (0.067) decrease in the revenue-per-visit difference between the MNL and SF-ML approaches.

Table 6. Mechanism Behind MNL-Based Model's Superior Financial Performance: Summary Statistics of Mechanisms

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

Notes. Standard errors are robust and clustered at the customer level. Reported are 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.

Table 7. Mechanism Behind MNL-Based Model's Superior Financial Performance: OLS Regression Results on Model Financial Performance

	Dependent variable:		
	Revenue (1)	PurchaseIncidence (2)	PricePerPurchase (3)
SF-ML	−0.987***	−0.004***	−6.349***
	(0.073)	(0.0001)	(1.820)
AF-ML	0.032	0.001***	−5.895***
	(0.077)	(0.0001)	(2.069)
Buyer controls	Yes	Yes	Yes
Seller fixed effect	Yes	Yes	Yes
Date fixed effect	Yes	Yes	Yes
Observations	10,410,587	10,410,587	237,417

Notes. Standard errors are robust and clustered at the customer level. Reported are the corresponding results from OLS regressions controlling for customer characteristics, seller fixed effects, and date fixed effects.

***$p < 0.01$

7. Conclusion

In this paper, to the best of our knowledge, we document the first full-scale implementation of a customer-choice-model-based product-recommendation system in Alibaba's product-recommendation setting. We find that although the simpler, more interpretable MNL model might be less accurate than a black-box machine-learning approach, it ultimately considers a more sophisticated assortment problem, which leads to gains in revenue exceeding 28% compared with its machine-learning counterpart. Although our work documents important empirical findings regarding the practical value of the MNL model in comparison with machine-learning-based models, it is not without its own limitations. First, our results are specific to the Alibaba coupon recommendation setting that we consider and, hence, cannot be extrapolated to other companies or other settings with certainty. That said, our findings do convey an important overarching message to practitioners—namely, that it is essential to consider the sophistication of both the estimation and assortment phases of any recommender system. Second, the results of Section 6.4 suggest that future work is needed to build choice models that are able to model click behavior, as well as customers buying multiple distinct items from a set of offered products. Third, our work has focused exclusively on the MNL model, leaving open the door for future work to explore the potential of more sophisticated choice models in product-recommendation settings and beyond. Finally, our focus is on implementing revenue-management algorithms in one important sales channel at Alibaba, and, hence, our approach does not account for the possibility of across-channel revenue cannibalization. Therefore, one potential future line of work is to study how to design and implement revenue-management algorithms on retailing platforms that consider across-channel cannibalization effects.

Table 8. Heterogeneous Treatment Effect

	Dependent variable:	
	Revenue	
	(1)	(2)
SF-ML	−0.822****	−0.780****
	(0.068)	(0.065)
SF-ML × MultiPurchaseCount	0.030***	
	(0.012)	
SF-ML × Click − to − Purchase		0.067*
		(0.035)
Customer controls	Yes	Yes
Seller fixed effects	No	No
Date fixed effects	Yes	Yes
Observations	5,326,664	5,326,664

Notes. Standard errors are robust and clustered at the customer level. This table reports the results based on Equation (8).

*$p < 0.10$; **$p < 0.05$; ****$p < 0001$.

Acknowledgments

Jacob Feldman and Dennis J. Zhang contributed equally and are ranked alphabetically.

Appendices

Appendix A. Estimation Case Study: Machine Learning vs. MNL

In this section, we present a case study in which we fit both MNL and machine-learning models to historical sales data generated in April 2018 from the coupon subpages of the 10 most popular sellers on Tmall.com. Because of the fact that we only use sales data from 10 sellers to fit our models, the scale of the estimation problems we consider is much smaller than the one encountered within the recommender systems we actually implement on Alibaba. Furthermore, because the exact nature of the machine-learning methods used by Alibaba must remain confidential, we are not able to replicate their methods or results exactly in this case study. Instead, we fit machine-learning models inspired by the current practice at Alibaba in the sense that both estimation schemes rely on gradient-boosted decision trees to estimate the click and purchase probabilities. It is important to note that the intent of this case study is not to perfectly replicate the estimation problem faced by Alibaba, but, instead, to show that it is relatively straightforward to fit machine-learning models that outperform the MNL fits in terms of predictions accuracy.

A.1. Top 10 Seller Statistics

Alibaba provided us with two weeks of historical sales data from the 10 sellers on Tmall.com that experienced the largest volume of traffic in April 2018. We note that this two-week time period does not overlap with the time

horizon of our field experiments. Table A.1 provides an extensive summary of the available sales data for each seller. Furthermore, for each arriving customer t and offered product $t \in S_t$, the feature vector $\mathbf{X_{jt}}$ gives the values of the 25 features with the highest importance scores according to the machine-learning approaches that have been utilized in the past. These are the same 25 features that are used within the two frameworks that were ultimately implemented.

A.2. Accuracy Metrics and Models Tested

For each seller, we randomly select 75% of its sales data to be used for fitting the models and hold out the remaining 25% of the data to test the accuracy of these models. After splitting the data in this way, we aggregate all of the training data from each seller into a single training set. This set-up most closely resembles the current practice at Alibaba, where the machine-learning models are fit to sales data aggregated across all sellers. Once the MNL and machine-learning models have been trained, we measure the accuracy of each fitted model using two metrics that are computed using the sales data exclusively from each seller's testing set restricted to customers who purchase exactly one item. In computing these accuracy metrics, we ignore customers who make multiple purchases, which has a negligible effect on our results because multiple products were purchased in approximately 0.01% of customer visits. That said, we defer explanations for why no-purchase events are ignored until the two accuracy metrics are formally defined, because this understanding will help elucidate our choice. The series of steps described above—75/25 train/test split, fitting the models, and computing the accuracy metrics on the test data set—make up what we refer to as a single trial. We eventually present the average accuracy metrics for each seller over 20 trials.

It is important to note that one potential metric that could be used to assess fitting accuracy is the log-likelihood on a hold-out sample of sales data. This metric is often referred to as the out-of-sample log-likelihood, and it has been a popular metric for assessing the accuracy of fitted customer-choice models in the revenue-management literature (see Şimşek and Topaloglu (2018), 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. To see this, consider the likelihood under both approaches for the arrival of a single customer who is offered assortment S and who purchases product $j \in S$. Under the MNL model, the likelihood is simply $P_j(S)$, which represents the probability that product j in purchased under assortment S. On the other hand, the likelihood under the machine-learning model is $p_j \cdot \prod_{i \in S: i \neq j}(1 - p_i)$, which corresponds to the likelihood that product j is purchased and all other offered products are not purchased. Hence, the likelihoods under the two fitted models inherently measure the probabilities of different events. As a result, we have chosen to use the following two metrics to measure prediction accuracy, which assesses how well the fitted models are able to predict the product that the arriving customer ultimately purchased.

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

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

Table A.1. Key Seller Statistics

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

Note. This table reports the key statistics, including categories, number of products and conversion rates, for the top 10 sellers that we use for this case study.

We fit the following two models. The code and tuning parameters for both models can be found in the Jupyter notebook titled "Implementation Details," which is included in the online appendix.

1. **The MNL choice model (MNL):** We fit this model by solving Problem (1) via Tensorflow implementation.

2. **The machine-learning models (Trees):** We use gradient-boosted classification trees to estimate the click probabilities $\mathbb{P}(c_{jt} = 1)$ and the conditional purchase probabilities $\mathbb{P}(z_{jt} = 1|c_{jt} = 1)$. More specifically, we use Catboost (Prokhorenkova et al. 2018), a novel gradient-boosting toolkit.

A.3. Results

The results for each seller with regards to the two accuracy metrics are presented in Table A.2. The first two columns identify the seller number and the fitted model. Columns (3) and (5) specify the mean classification accuracy and average rank, respectively, over 20 trials. Columns (4) and (6) correspond to the percentage improvement in performance of the machine-learning models over the standard MNL fits. The results in this table clearly show that a simple out-of-the-box machine-learning method is able to outperform the MNL model with regards to both accuracy metrics. Furthermore, in Figure A.1, we plot the percentage improvements of the machine-learning fits as a function of the purchase and click rate of each seller. Barring the outlier that is seller 6 (the points at the top of each plot), we observe that the percentage improvement is inversely related to both of these rates, indicating that the machine-learning models are better at handling sparse sales data.

Appendix B. Running-Time Improvements

In this section, we measure the improvement in running time that result from using Algorithm 1 in lieu of the approach proposed in Rusmevichientong et al. (2010). To do so, we first randomly generate a large test bed of MNL-based cardinality-constrained assortment instances. Then, for each instance, we carry out both algorithms and record their respective running times. Both approaches were implemented in Python 3.6 on a MacBook with a 1.1-GHz Dual-Core Intel Core processor and 8 GB of RAM. Our exact implementations can be found in the Jupyter notebook titled "Running Time Experiments" in the online appendix. Although this implementation does not exactly match the one carried out in practice, it still allows for fair comparison between the relative running times of both approaches.

B.1. Instance Generator

For each instance, we choose $C = 6$ and vary $n \in \{100, 500, 1000\}$ to mirror the structure of the Alibaba instance. The revenue of product is generated from a log-normal distribution with mean zero and scale parameter one. The preference weight of each product i is then set to be $w_i = e^{\alpha_i - \boldsymbol{\beta}_i \cdot r_i}$, where both α_i and $\boldsymbol{\beta}_i$ are drawn uniformly from the interval $[0, 1]$.

B.2. Results

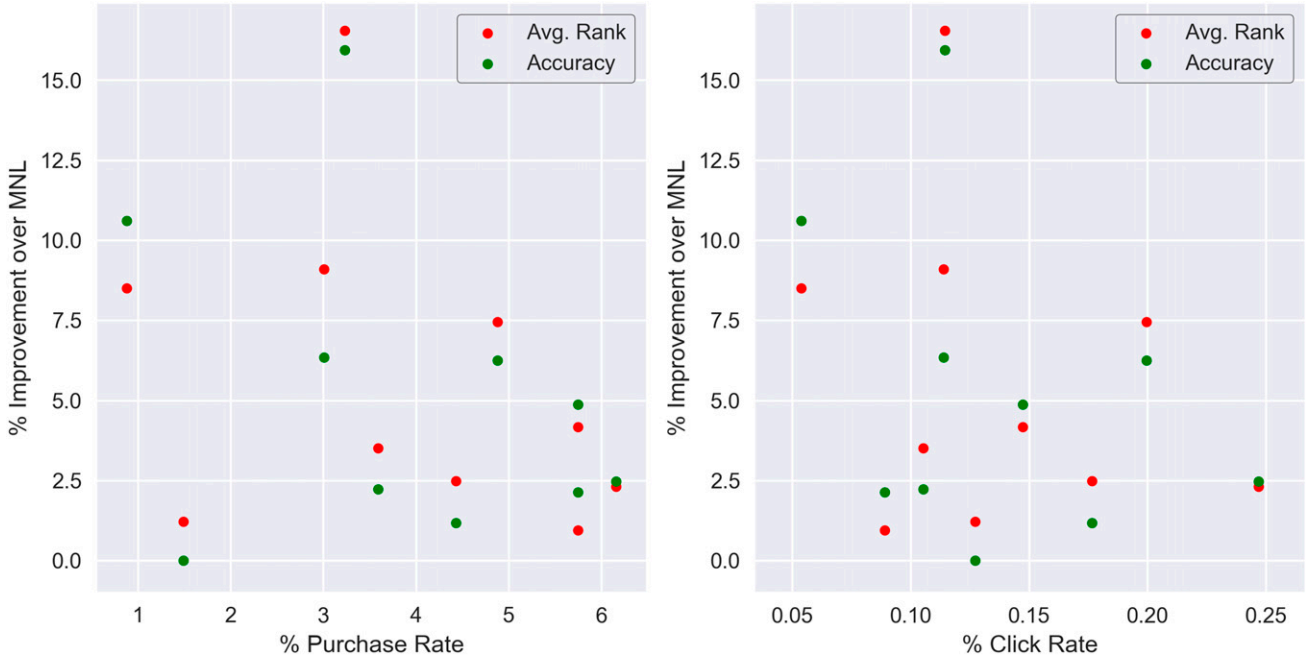
For each unique value of n, we generate 100 different instances. For each distinct instance, we record $\tau_{\text{new}}/\tau_{\text{old}}$, where τ_{new} and τ_{old} are the recorded running time of our approach and the approach of Rusmevichientong et al. (2010), respectively. Table B.1 gives the results of our experiments, where the second column reports the average ratio of the running times across the 100 trials. We observe that our new approach has the potential to be up to two or three times faster than the original approach. Moreover, as we would expect, the degree of improvement grows as n increases.

Table A.2. Predictive Performance of the Fitted Models

Seller number	Fitted model	Classification accuracy	Improvement over MNL	Average rank	Improvement over MNL
1	MNL	0.80	—	1.30	—
1	Trees	0.85	6.25%	1.21	7.44%
2	MNL	0.66	—	1.66	—
2	Trees	0.73	10.61%	1.53	8.50%
3	MNL	0.63	—	1.80	—
3	Trees	0.67	6.35%	1.65	9.10%
4	MNL	0.85	—	1.24	—
4	Trees	0.86	1.18%	1.21	2.48%
5	MNL	0.81	—	1.33	—
5	Trees	0.83	2.47%	1.30	2.31%
6	MNL	0.69	—	1.55	—
6	Trees	0.85	15.94%	1.33	16.54%
7	MNL	0.82	—	1.25	—
7	Trees	0.86	4.88%	1.20	4.17%
8	MNL	0.90	—	1.18	—
8	Trees	0.92	2.22%	1.14	3.51%
9	MNL	0.60	—	1.66	—
9	Trees	0.60	0%	1.64	1.21%
10	MNL	0.94	—	1.07	—
10	Trees	0.96	2.13%	1.06	0.94%

Note. This table shows the out-of-sample average classification accuracy and average rank of machine-learning and MNL models over each of the top 10 sellers.

Figure A.1. (Color online) The Percent Improvement of ML Fits over the MNL Fits as a Function of Purchase and Click Rates



Appendix C. Long-Term Impact of Our Field Experiment

In this brief section, we show that our field experiment did not have detrimental effects on future customer behavior. For this purpose, the upper panel of Figure C.1 gives the average GMV from July 1, 2018, to July 31, 2018 (four months after our experiments) of customers who were assigned the MNL group and the SF-ML group. The lower panel of this figure shows the fraction of these customers who made at one least one purchase on Tmall or Taobao during July 2018. We clearly observe that the purchasing statistics are statistically similar across both buckets of customers.

Appendix D. The Comparison of the Full-Feature MNL and the Machine-Learning Model

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

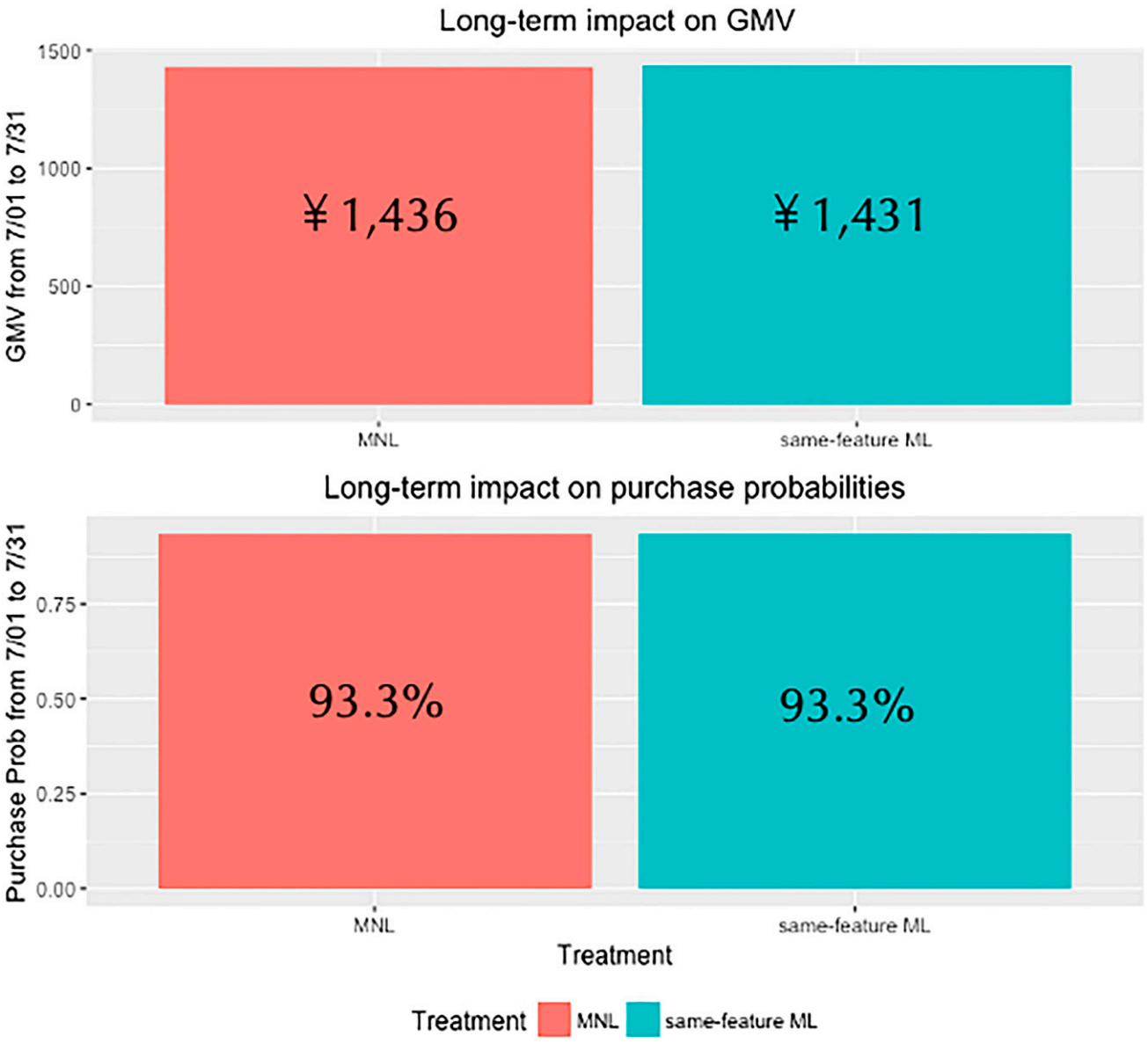
Over the five days of our experiment, we observed 3,591,021 customer arrivals from 2,247,663 million unique customers. A total of 1,125,381 of these customers were randomly assigned to the MNL-choice-model-based approach on all features (i.e., AF-MNL approach), while 1,122,282 were assigned to Alibaba's original machine-learning approach on all features (i.e., AF-ML approach). The customers under the AF-MNL approach collectively spent 15,114,748 RMB during the five days, while the customers in the AF-ML approach spent 14,621,580 RMB, an improvement of 493,168 RMB (i.e., 3.37% during the experimental period). Using the data from the previous experiment, we can estimate that this setting generates approximately $18 + 14 + 17.8 = 49.8$ million RMB per week from all customer visits. Therefore, we estimate that this 3.37% improvement corresponds to $49.8 \times 52 \times 3.37\% = 87.26$ million RMB (12.42 million USD) in one year.

Table D.1 presents the GMV per customer visit generated by these two approaches on all features. The first row shows that the AF-MNL and AF-ML approaches generate RMB 4.79 and 4.64 per customer visit, respectively, and the difference is 0.15 RMB per customer visit; in other words, the AF-MNL approach improves the revenue per customer visit by 3.37% compared with the AF-ML approach (p-value < 0.0001). This demonstrate that the

Table B.1. Running-Time Improvement Afforded by Algorithm 1

n	τ_{new}/τ_{old}
100	0.51
500	0.35
1,000	0.31

Figure C.1. (Color online) The Long-Term Impact of the MNL-Based Approach vs. the ML-Based Approach with Same Features



MNL-based approach outperforms the machine-learning-based approach, even if both approaches are utilizing all features. We note that the machine-learning-based approach on all features is exactly the algorithm and the feature set that are used by Alibaba to recommend products prior to our collaboration. This shows that our algorithm improves the state-of-art recommender system of Alibaba by 3.37%, which leads to the adoption of our algorithm as the main recommendation algorithm in this setting on Alibaba. We also note that this improvement based on all features (i.e., 3.37%) is more modest than the improvement based on only the top 25 features (i.e., 28.0%), which may demonstrate that

Table D.1. All-Feature Model Financial Performance

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

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

machine-learning-based approaches can more easily scale up to more features.

Endnotes

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

[2] The number of customers aggregated across two weeks surpassed Alibaba's limit on the amount of data that can be reported in a published research paper. This is the reason we only report the results for the first week of the experiments. Our results do not change qualitatively if we use the second week of data.

[3] 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 and Dusse 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.

[4] The unit of analysis in the randomization test of sellers' characteristics is at the visit level. In other words, the reported averages are taken across individual visits. We also conducted the same analysis at the seller level and found the results to be similar. The unit of analysis in the randomization test of customers' characteristics is at the customer level.

[5] One may worry that, by generating more pricey sales, the MNL-based approach will churn out customers in the treated group in the long-run. In Online Appendix 10, we consider a month-long period four months after our experiments and show that the average revenue generated by customers treated by the MNL-based approach matches the average revenue generated by customers assigned the SF-ML approach months after the experiment.

[6] Our results are qualitatively robust if the number-of-visits cutoffs are set to 500 or 1,000.

References

Breese JS, Heckerman D, Kadie C (1998) Empirical analysis of predictive algorithms for collaborative filtering. Cooper GF, Moral F, eds. *UAI'98 Proc. 14th Conf. Uncertainty Artificial Intelligence* (Morgan Kaufmann Publishers Inc., Burlington, MA), 43–52.

Cachon GP, Gallino S, Olivares M (2019) Does adding inventory increase sales? Evidence of a scarcity effect in US automobile dealerships. *Management Sci.* 65(4):1469–1485.

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

Caro F, Gallien J (2010) Inventory management of a fast-fashion retail network. *Oper. Res.* 58(2):257–273.

Chen X, Simchi-Levi D (2004) Coordinating inventory control and pricing strategies with random demand and fixed ordering cost: The finite horizon case. *Oper. Res.* 52(6):887–896.

Chen X, Pang Z, Pan L (2014) Coordinating inventory control and pricing strategies for perishable products. *Oper. Res.* 62(2): 284–300.

Covington P, Adams J, Sargin E (2016) Deep neural networks for YouTube recommendations. *Proc. 10th ACM Conf. Recommender Systems* (Association for Computing Machinery, New York), 191–198.

Cui Y, Duenyas I, Sahin O (2018) Pricing of conditional upgrades in the presence of strategic consumers. *Management Sci.* 64(7): 3208–3226.

Cui Y, Orhun AY, Duenyas I (2019) How price dispersion changes when upgrades are introduced: Theory and empirical evidence from the airline industry. *Management Sci.* 65(8):3835–3852.

Davidson J, Liebald B, Liu J, Nandy P, Van Vleet T, Gargi U, Gupta S, et al (2010) The YouTube video recommendation system. *Proc. Fourth ACM Conf. Recommender Systems* (Association for Computing Machinery, New York), 293–296.

Davis JM, Gallego G, Topaloglu H (2014) Assortment optimization under variants of the nested logit model. *Oper. Res.* 62(2):250–273.

Elmachtoub AN, Hamilton M (2017) The power of opaque products in pricing. Preprint, submitted August 28, https://dx.doi.org/10.2139/ssrn.3025944.

Ferreira KJ, Bin HAL, Simchi-Levi D (2015) Analytics for an online retailer: Demand forecasting and price optimization. *Manufacturing Service Oper. Management* 18(1):69–88.

Friedman JH (2002) Stochastic gradient boosting. *Comput. Statist. Data Anal.* 38(4):367–378.

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

Gallego G, Li A, Truong V-A, Wang X (2020) Approximation algorithms for product framing and pricing. *Oper. Res.* 68(1):134–160.

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

Jahrer M, Töscher A, Legenstein R (2010) Combining predictions for accurate recommender systems. *Proc. 16th ACM SIGKDD Internat. Conf. Knowledge Discovery Data Mining* (Association for Computing Machinery, New York), 693–702.

LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521(7553):436–444.

Li A, Talluri K (2019) An updated risk-ratio procedure in joint estimation of customer choice model parameters and arrival rate. Working paper, London School of Economics, London.

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

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

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

Naumov M, Mudigere D, Shi H-JM, Huang J, Sundaraman N, Park J, Wang X et al. (2019) Deep learning recommendation model for personalization and recommendation systems. Preprint, submitted May 31, https://arxiv.org/abs/1906.00091.

Netessine S, Rudi N (2003) Centralized and competitive inventory models with demand substitution. *Oper. Res.* 51(2):329–335.

Prokhorenkova L, Gusev G, Vorobev A, Dorogush AV, Gulin A (2018) Catboost: Unbiased boosting with categorical features. Bengio S, Wallach HM, Larochelle H, Grauman K, Cesa-Bianchi N, eds. *NIPS'18 Proc. 32nd Internat. Conf. Neural Inform. Processing Systems* (Curran Associates, Red Hook, NY), 6639–6649.

Ramachandran K, Tereyağoğlu N, Xia Y (2018) Multidimensional decision making in operations: An experimental investigation of joint pricing and quantity decisions. *Management Sci.* 64(12): 5544–5558.

Ravikumar P, Wainwright MJ, Lafferty JD (2010) High-dimensional Ising model selection using 1-regularized logistic regression. *Ann. Statist.* 38(3):1287–1319.

Ricci F, Rokach L, Shapira B (2011) Introduction to recommender systems handbook. Ricci F, Rokach L, Shapira B, Kantor PB, eds. *Recommender Systems Handbook* (Springer, Boston), 1–35.

Rivest R, Dusse S (1992) The md5 message-digest algorithm. Technical report, MIT Laboratory for Computer Science, Cambridge.

Rumyantsev S, Netessine S (2007) What can be learned from classical inventory models? A cross-industry exploratory investigation. *Manufacturing Service Oper. Management* 9(4):409–429.

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

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

Sumida M, Gallego G, Rusmevichientong P, Topaloglu H, Davis J (2019) Revenue-utility tradeoff in assortment optimization under the multinomial logit model with totally unimodular constraints. Tech. rep., Cornell University, Ithaca, NY.

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

Tereyağoğlu N, Fader PS, Veeraraghavan S (2017) Pricing theater seats: The value of price commitment and monotone discounting. *Production Oper. Management* 26(6):1056–1075.

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

Vulcano G, Abdullah T (2020) Demand estimation under the multinomial logit model from sales transaction data. *Manufacturing Service Oper. Management*, ePub ahead of print October 9, https://doi.org/10.1287/msom.2020.0878.

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

Jacob Feldman is an associate professor at Washington University's Olin Business School. His research interests lie at the intersection of approximation algorithms and revenue management.

Dennis J. Zhang is also an associate professor at Washington University's Olin Business School. His research mainly focuses on empirical operations management.

Xiaofei Liu is an engineer manager at Alibaba Corporation.

Nannan Zhang is an engineer at Alibaba Corporation.