Submitted to *Manufacturing & Service Operations Management*
manuscript (Please, provide the mansucript number!)

Demand Learning and Pricing for Varying Assortments

Kris Johnson Ferreira
Harvard Business School, Boston, MA 02163, kferreira@hbs.edu

Emily Mower
Harvard University, Cambridge, MA 02138, emower@g.harvard.edu

Problem Definition: We consider the problem of demand learning and pricing for retailers who offer assortments of substitutable products that change frequently, e.g., due to limited inventory, perishable or time-sensitive products, or the retailer's desire to frequently offer new styles.

Academic/Practical Relevance: We are one of the first to consider the demand learning and pricing problem for retailers who offer product assortments that change frequently, and we propose and implement a learn-then-earn algorithm for use in this setting. Our algorithm prioritizes a short learning phase, an important practical characteristic that is only considered by few other algorithms.

Methodology: We develop a novel demand learning and pricing algorithm that learns quickly in an environment with varying assortments and limited price changes by adapting the commonly used marketing technique of conjoint analysis to our setting. We partner with Zenrez, an e-commerce company that partners with fitness studios to sell excess capacity of fitness classes, to implement our algorithm in a controlled field experiment to evaluate its effectiveness in practice using the synthetic control method.

Results: Relative to a control group, our algorithm led to an expected initial dip in revenue during the learning phase, followed by a sustained and significant increase in average daily revenue of 14-18% throughout the earning phase, illustrating that our algorithmic contributions can make a significant impact in practice.

Managerial Implications: The theoretical benefit of demand learning and pricing algorithms is well understood – they allow retailers to optimally match supply and demand in the face of uncertain pre-season demand. However, most existing demand learning and pricing algorithms require substantial sales volume and the ability to change prices frequently for each product. Our work provides retailers who do not have this luxury a powerful demand learning and pricing algorithm that has been proven in practice.

1. Introduction

Within the last decade, demand learning and pricing has been at the forefront of academic research in the revenue management field, and numerous algorithms have been proposed (see Section 1.1). Most of these algorithms consider a single product - or in rarer cases, a single assortment of products - that is offered throughout a finite selling season, and the retailer can offer different prices for the product(s) to learn the demand at each price point and maximize total-season revenue. Furthermore, most of these algorithms require a high frequency and volume of price changes in order to reap the benefits of demand learning. Inherent in these two modeling choices are the assumptions that (*i*) the retailer has substantial sales volume of each product, and (*ii*) the retailer is able and willing to frequently change the price of each product. Interestingly, despite the plethora of demand learning and pricing algorithms in the academic literature, non-promotion based retail price changes were estimated to have occurred only once every 3.7 months between 2014-2017 (Cavallo (2018)), signaling that, in fact, many retailers have not adopted such algorithms.

We are motivated by conversations that we have had with numerous retailers who have yet to implement demand learning and pricing algorithms. Although these retailers understand their potential benefits, many of these retailers share similar sentiments. First, many products they sell do not have substantial sales volume to reap the benefits of dynamic pricing. This appears to be particularly common for retailers who offer assortments of products that change frequently, e.g., due to limited inventory, perishable or time-sensitive products, or simply the retailer's desire to frequently offer new styles; such settings are becoming more and more prevalent, especially in e-commerce (e.g., Petro (2018)). Second, many retailers are unable or unwilling to change prices multiple times within a single assortment due to technological constraints or concerns about negative customer perception or strategic consumer behavior (e.g., Garbarino and Lee (2003) and Haws and Bearden (2006)). Some retailers have suggested that changing prices only at the beginning of each new assortment - rather than changing prices within an assortment - is desirable because their technology easily supports such price changes and/or they believe that timing price changes with assortment changes would likely be better received by the customer.

With these characteristics in mind, we model customer demand using a contextual, attribute-based multinomial logit (MNL) choice model and develop a demand learning and pricing algorithm for the purpose of quickly learning consumer demand with minimal price changes to maximize revenue when product assortments may change over time. Our algorithm sets a price for each product at the beginning of each assortment which is offered to all customers who shop that assortment. The retailer observes each customer's purchase decision and can use this information when choosing the prices for subsequent assortments. In these retail settings, it is necessary to carry over any demand learning occurring from one assortment to the next assortment, and to do so, we assume that products can be fully characterized by a set of attributes; thus demand learning occurs at the attribute-level as opposed to the product-level, allowing learning to be transferred to newly offered products that share attributes with products from previous assortments.

Our algorithm follows a learn-then-earn approach, where at first the retailer chooses prices to learn demand as quickly as possible, and then after the retailer is sufficiently confident in the estimated demand model, the retailer prices to earn, with the goal of maximizing revenue (equivalently, profit). During the pricing to learn phase, we adapt techniques from conjoint analysis - a method that is common in the marketing literature and most often used by brands to help make product design decisions - to offer prices that maximize the expected information gain in each assortment, ultimately learning the parameters of the demand model as quickly as possible. During the pricing to earn phase, we price in a greedy fashion by assuming that the current parameter estimates are the true parameters and maximizing revenue under this assumption. Regardless of the phase, at the end of each assortment our algorithm updates the demand model parameter estimates with the current assortment's observed purchase data.

For the development and implementation of our work, we collaborated with Zenrez, an e-commerce company that partners with fitness studios across the United States and Canada to sell excess capacity of fitness studio classes. Every night at 9:00pm, Zenrez posts classes (i.e., products) that have remaining capacity for the following day and offers them at a discounted price via a widget located on the partner studios' webpages or app. When a user views the widget, they see all next-day classes offered by Zenrez for that fitness studio; see Figure 1 for an example. Each class is characterized by features such as class type (e.g., yoga, spin, etc.), time of day, and duration. The assortment of classes changes each day, and prices can vary across assortments but are fixed within each assortment in order to avoid negative customer perception. Zenrez has the flexibility to choose an integer price for each class within a studio-specified interval and earns a commission proportional to the selling price for each class sold via their widget.

To implement our demand learning and pricing algorithm, we developed a fully-automated pricing tool at Zenrez. It is run automatically every day for a given studio, setting prices for the following day's assortment of classes. To evaluate the effectiveness of our algorithm, we conducted a three-month controlled field experiment where prices for studios in the treatment group were set according to our algorithm while prices for studios in the control group were set according to Zenrez's baseline pricing policies, and we used synthetic controls (Abadie et al. 2010) to estimate the treatment effect: percent increase in average daily revenue between studios in the treatment group compared to the control group. Relative to studios in the control group, our algorithm led to an 8.5% increase in average daily revenue over the three-month experiment. The average effect on revenue balances an expected initial dip in revenue experienced when pricing to learn followed

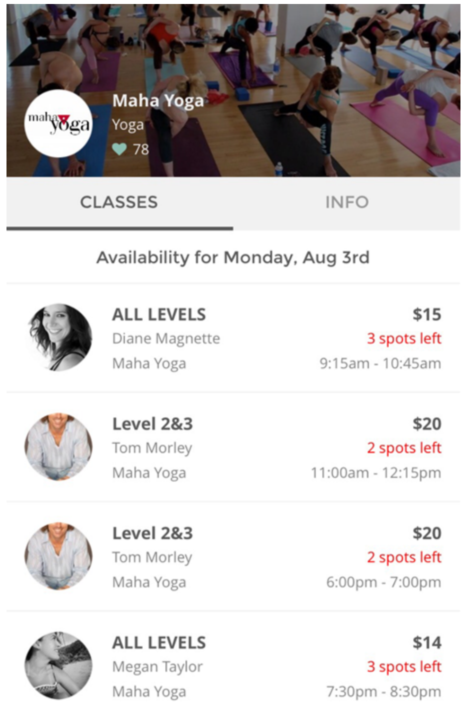

Figure 1 Example of Zenrez's widget, showing the assortment of classes offered for partner studio Maha Yoga on Monday, August 3rd. Retrieved from More Than Mary, 2015, http://www.morethanmary.com/2015/08/03/new-fitness-apps/.

by a significant increase in revenue (14-18%) when pricing to earn; it is reasonable to expect that gains of similar magnitudes would endure over future periods if the algorithm were run for longer.

Our main contributions in this paper are three-fold. First, we contribute to the nascent revenue management literature for this prevalent retail setting where product assortments change frequently, e.g., due to limited inventory, perishable or time-sensitive products, or the desire to frequently offer new styles. To account for many retailers' desire to not change prices multiple times within a single assortment, we use the beginning of each assortment change as a natural time in which to change prices. We know of only one other paper on demand learning and pricing for the multi-product, discrete choice setting that also accounts for varying assortments (Javanmard et al. (2020)); we compare our work with Javanmard et al. (2020) in Section 1.1 and present simulations comparing our algorithm's performance with theirs in Section 3.1. We hope that our work encourages other researchers to consider operations and marketing problems targeted for retailers with frequent assortment changes, limited sales volume, and/or an interest in limited price changes.

Second, we develop a novel learn-then-earn algorithm - *Fast Learning and Pricing for Varying Assortments* - that learns quickly in an environment with varying assortments and limited price changes by adapting the commonly used marketing technique of conjoint analysis to our setting. A short learning phase is particularly important in our setting where retailers want to minimize negative customer perception from volatile price experimentation, as well as for retailers who have limited sales volume or a short selling season and thus cannot afford lengthy experimentation. Furthermore, when external factors change that may influence demand model parameters, a short learning phase allows retailers to restart the algorithm and quickly learn the new demand model parameters. Most demand learning and pricing algorithms proposed in the academic literature do not consider a short learning phase as an algorithmic characteristic and instead focus on asymptotic analyses. We are one of the first to consider such an algorithmic characteristic in this retail setting, as well as the first to employ conjoint analysis in demand learning and pricing. We hope that our work encourages other researchers to consider the length of the learning phase and/or conjoint analysis when developing learning and decision-making algorithms.

For our third contribution, we partner with Zenrez to estimate the effectiveness of our algorithm in a controlled field experiment, and we illustrate how to use synthetic controls to estimate the treatment effect, a popular method recently proposed to evaluate policy interventions. The results of our field experiment illustrate that our algorithmic contributions can make a significant impact in practice, and we hope that our work will inspire other retailers to implement our demand learning and pricing algorithm. To the best of our knowledge, ours is the first paper to present the implementation of a multi-product demand learning and pricing algorithm in practice.

1.1. Literature Review

Our paper contributes to the vast literature on demand learning and pricing. For a more in-depth review of the relevant literature, we refer the reader to the extensive survey by den Boer (2015); examples of more recent papers include Ban and Keskin (2020), Ferreira et al. (2018), and Misra et al. (2019). The tension that motivates demand learning and pricing research is the classic exploration-exploitation trade-off, which requires the retailer to learn customer demand in order to identify revenue maximizing prices while minimizing revenue lost to pricing-to-learn rather than pricing-to-earn. Our model and algorithm are differentiated from most others presented in this literature in three critical ways: (*i*) we consider a multi-product, discrete choice setting with varying product assortments, (*ii*) we prioritize a short learning phase so that the retailer only engages in limited price experimentation, and (*iii*) we implement and evaluate our algorithm in practice.

Multi-product, Discrete Choice with Varying Assortments: First, we consider a multi-product, discrete choice setting where the assortment of substitutable products offered to customers changes over time. This setting necessitates modeling product attributes (context) and learning attribute-level - as opposed to product-level - demand parameters in order to transfer learning to newly offered products in each assortment. Most multi-product demand learning and pricing algorithms are not contextual and thus cannot be applied to settings like ours where the assortment changes frequently. Javanmard et al. (2020) is the only other work in this setting that utilizes a parametric demand model; in fact, we use an identical utility model with heterogeneous price sensitivities and customer demand described by a multinomial logit (MNL) model. That said, there are some key differences in the retail setting and motivation of our works that lead to different algorithmic contributions. In particular, Javanmard et al. (2020) are motivated by retailers who can change prices for *every* customer arrival whereas we are motivated by retailers who want to employ minimal price changes. In addition, Javanmard et al. (2020) allows prices to be selected from $\mathbb{R}^+$ (i.e., a continuous and unconstrained set), whereas we require prices to be selected from a finite set, a common practice among many retailers.

Regarding algorithmic contributions, Javanmard et al. (2020) propose an algorithm (M3P) that alternates between learning and earning phases, where in the learning phase it selects random prices, and in the earning phase it selects prices that maximize revenue based on the parameter estimates derived from the learning phases. Our algorithm differs in three respects: (*i*) we follow a learn-then-earn approach, which results in fewer periods where the retailer needs to engage in price exploration, (*ii*) during our learning phase, we adapt techniques from conjoint analysis to select prices that maximize the expected information gain in order to learn quickly, and (*iii*) during our earning phase, we continue to passively learn by updating our parameter estimates with new observations. Finally, Javanmard et al. (2020) evaluate their algorithm via an asymptotic analysis where T grows very large and show strong T-period regret, whereas we implement our algorithm in practice (small, finite T) and evaluate its performance in a controlled field experiment. As a comparison, we present the performance of both algorithms via numerical simulations in Section 3.1.

Miao and Chao (2020) also consider a parametric, demand learning and pricing problem for the multi-product, discrete choice setting where the assortment of substitutable products offered to the customer can change over time, selected from a static set of N products in each period. They also use an MNL demand model but use product-specific mean utilities and price sensitivities as opposed to attribute-specific.

Very recently, the multinomial logit contextual bandit model was introduced in Oh and Iyengar (2019a); Oh and Iyengar (2019a), Oh and Iyengar (2019b), and Chen et al. (2020) study a demand learning and assortment optimization problem under the MNL contextual bandit model. Our model can be cast as an MNL contextual bandit model; our contribution related to these recent papers is that we study price as opposed to assortment optimization. They develop upper confidence bound and Thompson sampling based policies that utilize Fisher's information; we also use Fisher's information in the learning phase of our pricing algorithm.

Focus on Short Learning Phase: Most demand learning and pricing papers do not consider a short learning phase as an algorithmic characteristic and instead focus solely on asymptotic analyses. In fact, despite the attractiveness of limited price changes in practice, we know of only three other papers that consider such a characteristic in a demand learning and pricing algorithm– Cheung et al. (2017), Chen and Chao (2019), and Perakis and Singhvi (2020)–each of which consider a single product setting as opposed to our multi-product setting. Cheung et al. (2017) and Chen and Chao (2019) constrain the number or timing of price changes, whereas Perakis and Singhvi (2020) incorporate the desire for limited price changes when learning non-parametric demand.

Similar to Perakis and Singhvi (2020), we also do not impose a constraint on the number of price changes and instead incorporate the desire for limited price experimentation in our algorithmic design. This led us to utilize components of conjoint analysis in the learning phase of our algorithm, and our paper is the first to integrate conjoint analysis with demand learning and pricing, increasing the velocity at which learning can occur. Conjoint analysis is most commonly used to construct surveys or choice experiments that inform product design decisions. A subset of the broader field is devoted to choice-based conjoint analysis, often assuming MNL demand. In choice-based conjoint analysis, products are characterized by a set of attributes and the researcher constructs a hypothetical choice set of substitutable products with variation in their attributes. The researcher then asks respondents to select their favorite from among the set of substitutable (hypothetical) products. Using observed choices, the researcher can then estimate customer utility for each of the product attributes. Refer to Raghavarao et al. (2010) for more details.

To select the hypothetical choice sets used in these choice experiments, choice-based conjoint analysis uses principles of optimal experimental design to maximize information gain as measured by the determinant of the Fisher information matrix. One key challenge in doing so is that the optimal design is a function of the unknown parameters of the utility model. To address this challenge, marketers historically initialized the parameters at zero. Huber and Zwerina (1996) were the first to improve design efficiency by introducing a pre-experiment, which can be used to initialize parameters at more meaningful values. Sandor and Wedel (2001) built upon their work by using a Bayesian approach that incorporates managers' priors and accounts for their uncertainty. Later work by Sandor and Wedel (2005) demonstrated significant value in being able to observe choice behavior across several distinct choice experiments. We extend these ideas to our demand learning and pricing setting by considering each assortment to be a 'choice set' and allowing the algorithm to choose only the value of the price attribute for each product in the choice set; all other attributes are selected exogenously by the firm. Our algorithm is able to incorporate prior information when it is available and updates utility model parameter estimates after each choice set. In this way, we are able to sequentially observe choices across a range of choice sets and update our parameter estimates to reflect observed demand. Although our algorithm is sequential in nature, it differs from what are known as "adaptive designs" in the conjoint analysis literature, which change the designs for a single respondent based on his or her previous responses (e.g., Sauré and Vielma (2018)); this work would be more appropriate to use for a personalized pricing application.

Price Optimization Field Experiments: Despite the growing number of academic papers presenting new pricing algorithms, there have been very few documenting the implementation and validation of pricing algorithms in practice via field experiments. Caro and Gallien (2012) develop and implement a markdown price optimization tool at Zara. Besbes et al. (2020) develop a price optimization tool for rotable spare parts and evaluate its effectiveness at an aircraft OEM. Both of these papers dynamically change the price of products during the season but do not consider varying assortments, and both use a demand model where each product's demand is independent of the other products in the assortment. Fisher et al. (2018) address price optimization in light of competitor pricing and conduct a field experiment to evaluate their algorithm at Yihaodian; they consider a discrete choice setting, as do we, but they do not consider varying assortments. Ferreira et al. (2016) develop a price optimization tool for Rue La La that considers both varying assortments and the multi-product demand setting, albeit not discrete choice. Most importantly, none of these papers consider demand learning - and therefore the exploration-exploitation trade-off - in their algorithm or implementation. Thus for each case, the objective of their algorithm is to maximize revenue or profit, whereas our algorithm balances the exploration-exploitation trade-off by first maximizing information gain and subsequently maximizing revenue. We are aware of only one other paper that implements a demand learning and pricing algorithm in practice, Cheung et al. (2017), which consider a single-product setting without varying assortments.

2. Model

We consider a retailer who sells a varying assortment of substitutable products to customers over a selling season of length T. We are motivated by settings where T is relatively small–$\mathcal{O}(10)$ or $\mathcal{O}(100)$–compared to most demand learning and pricing models in the literature which study the asymptotic setting where T grows very large. In each time period $t = 1, ..., T$, the retailer offers a set of N_t products; each product can be fully characterized by an observable and exogenousvector of d features (attributes), $\boldsymbol{x_{it}} \in \mathbb{R}^d$ for product $i \in \{1, ..., N_t\}$, where $\boldsymbol{x_{it}} = \{x_{1it}, ..., x_{dit}\}$. We allow for a no purchase (outside) option encoded as product $i = 0$ for each assortment with d-dimensional vector $\boldsymbol{x_{0t}} = \vec{0}\ \forall t$. The feature vectors of products offered in different periods vary and thus we use "period t" and "assortment t" interchangeably. For each assortment t, the retailer selects price vector $\boldsymbol{p_t} = \{p_{0t}, p_{1t}, ..., p_{it}, ..., p_{N_t t}\}$ where the price p_{it} for product i is selected from a finite, discrete set of possible prices $\mathcal{P}_{it}$, a common practice for many retailers; for ease of notation, we define

$\mathcal{P}_t$ to be the finite set of all possible price vectors $\boldsymbol{p_t}$ which simply includes every combination of possible prices for each product. For the outside option $i=0$, we define $p_{0t}=0\ \forall t$ (or equivalently, $\mathcal{P}_{0t}=\{0\}$) without loss of generality and for notational convenience. Note that prices change from period to period, but not within a single period; this reflects many retailers' interests of changing prices only when assortments change as opposed to dynamically changing prices within a fixed assortment of products. Note that although our model and algorithm are motivated by the setting where prices do not change for a given assortment, allowing for such changes could be incorporated in our model formulation. For example, if a retailer was willing and able to change the price vector for an assortment three times, the assortment could be allocated three periods in our model, each offering the same set of products having identical feature vectors. Such a formulation would be particularly useful for retailers who - although they may not conduct frequent assortment changes - prefer only changing prices within an assortment a few times, yet still want to reap the benefits of demand learning and pricing.

We assume zero cost per product without loss of generality and for ease of exposition; note that this equates revenue to profit. We also assume that demand can be met in each period. Unless otherwise noted, consider all vectors to be column vectors. For convenience, we occasionally use the notation $(\boldsymbol{a},\boldsymbol{b})$ for vectors $\boldsymbol{a}\in\mathbb{R}^d$ and $\boldsymbol{b}\in\mathbb{R}^d$ to denote a column vector of length $2d$ where the first d entries are $\boldsymbol{a}$ and the second d entries are $\boldsymbol{b}$.

For customer j arriving in period t, we model her (random) utility from purchasing product i as a linear function of its features and price, specifically

$$u_{ijt}=\boldsymbol{x_{it}}^{\intercal}\boldsymbol{\beta^f}-\boldsymbol{x_{it}}^{\intercal}\boldsymbol{\beta^p}p_{it}+\epsilon_{ijt}\ . \tag{1}$$

Here, $\boldsymbol{\beta^f}$ and $\boldsymbol{\beta^p}$ are fixed parameters that are *unknown* to the retailer at the beginning of the season but can be learned throughout the season via observing purchase data; $\boldsymbol{\beta^f}\in\mathbb{R}^d$ reflects the impact of features on utility whereas $\boldsymbol{\beta^p}\in\mathbb{R}^d$ incorporates feature-specific price sensitivities. Since model parameters are defined with respect to features - as opposed to products - learning the value of these parameters from observing sales of one product can benefit new products that share some of the same features. Note that $\boldsymbol{x_{it}}^{\intercal}\boldsymbol{\beta^p}$ is the price sensitivity of product i and thus our model allows for heterogeneous price sensitivities across products. We let $\epsilon_{ijt}\ \forall i,j,t$ be a random component of utility and assume that it is drawn independently and identically from a standard Gumbel distribution. We assume the customer purchases the product (including outside option) that gives her the largest utility.

We note that our utility model is one specification of the well-known multinomial logit (MNL) discrete choice model that has been widely used in academia and in practice (see, e.g., Talluri and van Ryzin (2005) and Elshiewy et al. (2017)) that includes product features; it is identical to the utility model employed in Javanmard et al. (2020). The MNL model yields the following expression for the probability of a customer purchasing product i when offered assortment t for all $t = 1, ..., T$ and $i = 0, ..., N_t$, which we will refer to as the *demand model*:

$$q_{it} = \frac{\exp(\boldsymbol{x_{it}}^\intercal \boldsymbol{\beta^f} - \boldsymbol{x_{it}}^\intercal \boldsymbol{\beta^p} p_{it})}{\sum_{l=0}^{N_t} \exp(\boldsymbol{x_{lt}}^\intercal \boldsymbol{\beta^f} - \boldsymbol{x_{lt}}^\intercal \boldsymbol{\beta^p} p_{lt})} \ . \tag{2}$$

We define $\boldsymbol{q_t} = \{q_{0t}, q_{1t}, ..., q_{it}, ..., q_{N_t t}\}$.

For each period t, let M_t be a Poisson random variable with (unknown) arrival rate λ representing the number of customer arrivals in that period, i.e., the number of customers who shop assortment t. We assume that M_t is independent and identically distributed across periods, and that demand is independent across customers and periods; for many retailers such as those offering perishable or time-sensitive products, this independence assumption is naturally satisfied. Further define Y_{it} as a random variable representing the number of customers who purchase product i in assortment t, and note that $\sum_{i=0}^{N_t} Y_{it} = M_t$. At the end of period t, the retailer observes the quantity of each item purchased (the realizations of Y_{it} $\forall i = 0, 1, ..., N_t$), which we denote $\boldsymbol{y_t} = \{y_{0t}, y_{1t}, ..., y_{N_t t}\}$ where y_{it} is the quantity of product i purchased in period t, and y_{0t} is the number of customers who do not purchase any items. We use $m_t = \sum_{i=0}^{N_t} y_{it}$ to be the total number of customers who arrive in period t, i.e., the realization of random variable M_t. Note that we are assuming that the retailer can observe the number of customers who view the assortment and choose not to purchase, which is realistic for many online retailers like Zenrez via tracking clickstream data; although not as precise in the offline world, some brick-and-mortar retailers can estimate this using traffic patterns to the store.

The problem faced by the retailer is to design a non-anticipatory algorithm that selects a price vector $\boldsymbol{p_t}$ for each assortment $t = 1, ..., T$ in order to maximize the total revenue over the season. To be specific, "non-anticipatory" refers to the restriction that the algorithm can use only prior periods' price, feature, and observed sales information $(\boldsymbol{p_{t'}}, \boldsymbol{x_{it'}} \forall i \in \{1, ..., N_{t'}\}, \boldsymbol{y_{t'}} \quad \forall t' < t)$ when selecting a price vector for assortment t. This observed sales information enables the retailer to learn the arrival rate λ and demand model parameters $\boldsymbol{\beta^f}$ and $\boldsymbol{\beta^p}$, which in turn enables the retailer to learn demand not only for the offered price and product, but also for other prices and for other products that share some of the same features. Given our assumptions, we can write our objective considering only a single customer in each period and maximizing the expected revenue earned from that customer:

$$\max_{\boldsymbol{p_1} \in \mathcal{P}_1, ..., \boldsymbol{p_T} \in \mathcal{P}_T} \ \sum_{t=1}^{T} \boldsymbol{q_t}^\intercal \boldsymbol{p_t} \ . \tag{3}$$

Although not explicitly included in the objective, our algorithm is inspired by many retailers' desire to limit price exploration, i.e., how long it takes to learn parameters $\boldsymbol{\beta^f}$ and $\boldsymbol{\beta^p}$.

To summarize the sequence of events in our model, for each assortment t,

1. The retailer selects price vector $\boldsymbol{p_t} \in \mathcal{P}_t$ and offers N_t products with feature vectors $\boldsymbol{x_{it}}$ for $i = 1, ..., N_t$ and price vector $\boldsymbol{p_t}$ to all customers arriving during period t.
2. Each customer purchases at most one item.
3. The retailer observes $\boldsymbol{y_t}$ and can use this observation along with all prior sales information $(\boldsymbol{p_{t'}}, \boldsymbol{x_{it'}} \forall i \in \{1, ..., N_{t'}\}, \boldsymbol{y_{t'}} \quad \forall t' < t)$ when choosing the price vectors for subsequent assortments.

3. Fast Learning and Pricing for Varying Assortments

In this section, we propose an algorithm - *Fast Learning and Pricing for Varying Assortments* - to prescribe price vector $\boldsymbol{p_t}$ in each period t after observing purchase behavior in the prior period. Our algorithm follows a learn-then-earn approach, where at first the retailer chooses prices to learn demand as quickly as possible, and then after the retailer is sufficiently confident in the estimated demand model, the retailer prices to earn, with the goal of maximizing revenue. Algorithm 1 formally outlines our *Fast Learning and Pricing for Varying Assortments* algorithm. We start by initializing parameters $\boldsymbol{\hat{\beta}^f_1}$ and $\boldsymbol{\hat{\beta}^p_1}$ to $\vec{\mathbf{0}}$ and initializing $\hat{\lambda} > 0$ to reflect the retailer's prior belief for the arrival rate λ; note that these initializations are only relevant for pricing decisions made in period $t = 1$. The parameters $\hat{\lambda}$, $\boldsymbol{\hat{\beta}^f_t}$ and $\boldsymbol{\hat{\beta}^p_t}$ $\forall t = 1, ..., T$ will be used as empirical estimates of the true parameters λ, $\boldsymbol{\beta}^f$ and $\boldsymbol{\beta}^p$; in practice, the retailer could include prior information in the initializations if available. Using similar notation, we will let $\hat{q}_{it}$ be the purchase probability as defined in Equation (2) and using current parameter estimates $\boldsymbol{\hat{\beta}^f_t}$ and $\boldsymbol{\hat{\beta}^p_t}$. We initialize our pricing phase as *pricing to learn* and specify a Boolean switching criteria test that will dictate when the pricing phase switches to *pricing to earn.*

During the *pricing to learn* phase, our algorithm offers prices that maximize the expected information gain in each period in order to learn the parameters of the demand model as quickly as possible. Specifically, in *pricing to learn* period t, Algorithm 1 selects the price vector $\boldsymbol{p_t}$ from the set of feasible price vectors $\mathcal{P}_t$ that maximizes the determinant of the Fisher information matrix, which is the most common measure of information gain used in choice-based conjoint analysis. In our application, Fisher information in period t is a measure of how much information we can obtain about the unknown parameters $(\boldsymbol{\beta^f}, \boldsymbol{\beta^p})$ from a sample of customer purchases, $\boldsymbol{y_t}$, drawn from random variables Y_{it} $\forall i = 0, 1, ..., N_t$. Customer purchases are a function of the selected price vector $\boldsymbol{p_t}$, highlighting the impact of the pricing decision on how much information one can learn.

Algorithm 1: Fast Learning and Pricing for Varying Assortments

Input: Boolean switching criteria test, $SWITCH$;

Initialize parameters: $\hat{\boldsymbol{\beta}}^{\boldsymbol{f}}_{\boldsymbol{1}} = \hat{\boldsymbol{\beta}}^{\boldsymbol{p}}_{\boldsymbol{1}} = \vec{0}$ and $\hat{\lambda} > 0$;

Initialize pricing phase = *pricing to learn*;

for $t = 1, ..., T$ **do**

  if pricing phase = *pricing to learn* **then**

    Define $\boldsymbol{z_{is}} = (\boldsymbol{x_{is}}, -\boldsymbol{x_{is}} p_{is})\ \forall s \leq t$. Offer price vector

    $\boldsymbol{p_t^*} = \arg\max_{\boldsymbol{p_t} \in \mathcal{P}_t} \det \left[\sum_{s=1}^{t-1} m_s \sum_{i=0}^{N_s} \hat{q}_{is} \left(\boldsymbol{z_{is}} - \sum_{l=0}^{N_s} \hat{q}_{ls} \boldsymbol{z_{ls}} \right) \left(\boldsymbol{z_{is}} - \sum_{l=0}^{N_s} \hat{q}_{ls} \boldsymbol{z_{ls}} \right)^\top + \hat{\lambda} \sum_{i=0}^{N_t} \hat{q}_{it} \left(\boldsymbol{z_{it}} - \sum_{l=0}^{N_t} \hat{q}_{lt} \boldsymbol{z_{lt}} \right) \left(\boldsymbol{z_{it}} - \sum_{l=0}^{N_t} \hat{q}_{lt} \boldsymbol{z_{lt}} \right)^\top \right]$;

  end

  if pricing phase = *pricing to earn* **then**

    Offer price vector $\boldsymbol{p_t^*} = \arg\max_{\boldsymbol{p_t} \in \mathcal{P}_t} \sum_{i=0}^{N_t} p_{it} \cdot \frac{\exp(\boldsymbol{x_{it}^\top} \hat{\boldsymbol{\beta}}^{\boldsymbol{f}}_{\boldsymbol{t}} - \boldsymbol{x_{it}^\top} \hat{\boldsymbol{\beta}}^{\boldsymbol{p}}_{\boldsymbol{t}} p_{it})}{\sum_{l=0}^{N_t} \exp(\boldsymbol{x_{lt}^\top} \hat{\boldsymbol{\beta}}^{\boldsymbol{f}}_{\boldsymbol{t}} - \boldsymbol{x_{lt}^\top} \hat{\boldsymbol{\beta}}^{\boldsymbol{p}}_{\boldsymbol{t}} p_{lt})}$;

  end

  Observe demand $\boldsymbol{y_t}$;

  Estimate $(\hat{\boldsymbol{\beta}}^{\boldsymbol{f}}_{\boldsymbol{t+1}}, \hat{\boldsymbol{\beta}}^{\boldsymbol{p}}_{\boldsymbol{t+1}})$ using data observed through period t, i.e.,
  $(\boldsymbol{p_s^*}, \boldsymbol{x_{is}} \forall i \in \{1, ..., N_s\}, \boldsymbol{y_s}\ \ \forall s \leq t)$:
  $(\hat{\boldsymbol{\beta}}^{\boldsymbol{f}}_{\boldsymbol{t+1}}, \hat{\boldsymbol{\beta}}^{\boldsymbol{p}}_{\boldsymbol{t+1}}) = \arg\max_{\boldsymbol{\beta^f}, \boldsymbol{\beta^p}} \sum_{s=1}^{t} \sum_{i=0}^{N_s} y_{is} \ln \left(\frac{\exp(\boldsymbol{x_{is}^\top} \boldsymbol{\beta^f} - \boldsymbol{x_{is}^\top} \boldsymbol{\beta^p} p_{is}^*)}{\sum_{l=0}^{N_s} \exp(\boldsymbol{x_{ls}^\top} \boldsymbol{\beta^f} - \boldsymbol{x_{ls}^\top} \boldsymbol{\beta^p} p_{ls}^*)} \right)$;

  Estimate $\hat{\lambda}$ using data observed through period t: $\hat{\lambda} = \frac{1}{t} \sum_{s=1}^{t} m_s$;

  if pricing phase = *pricing to learn* and $SWITCH$ = TRUE **then**

    pricing phase = *pricing to earn*;

  end

end

Proposition 1: The Fisher information matrix for $(\boldsymbol{\beta^f}, \boldsymbol{\beta^p})$ for the multinomial choice demand model presented in (2) for τ periods is

$$I(\boldsymbol{\beta^f}, \boldsymbol{\beta^p}) = \sum_{t=1}^{\tau} m_t \sum_{i=0}^{N_t} \left[q_{it} \left((\boldsymbol{x_{it}}, -\boldsymbol{x_{it}} p_{it}) - \sum_{l=0}^{N_t} q_{lt} (\boldsymbol{x_{lt}}, -\boldsymbol{x_{lt}} p_{lt}) \right) \left((\boldsymbol{x_{it}}, -\boldsymbol{x_{it}} p_{it}) - \sum_{l=0}^{N_t} q_{lt} (\boldsymbol{x_{lt}}, -\boldsymbol{x_{lt}} p_{lt}) \right)^\top \right]. \tag{4}$$

The proof of Proposition 1 is presented in Appendix A. The inner summation characterizes the Fisher information matrix for a single period t, and the outer summation considers all τ periods, weighted by the number of customers who arrived in each period, m_t. Note that calculating the Fisher information matrix requires evaluating the demand model (2), which in turn requires knowledge of parameters $(\boldsymbol{\beta^f}, \boldsymbol{\beta^p})$; since these are unknown, Algorithm 1 uses their current estimated values, $(\hat{\boldsymbol{\beta}}^{\boldsymbol{f}}_{\boldsymbol{t}}, \hat{\boldsymbol{\beta}}^{\boldsymbol{p}}_{\boldsymbol{t}})$, to replace the probability that a customer purchases product i in period t, q_{it}, with its estimated probability, $\hat{q}_{it}$. In addition, since the number of customers arriving in period t, m_t, is unknown at the beginning of the period, we replace m_t with its expected value, $\mathbb{E}[M_t] = \hat{\lambda}$, when selecting the price for period t. We note that $\hat{\lambda}$ is simply the sample mean of the number of customers who arrived in previous periods, since $M_t \sim \text{Poisson}(\lambda)$. Analogous to D-optimal designs in choice-based conjoint analysis, we select prices to learn in Algorithm 1 by choosing the price vector which maximizes the determinant of the Fisher information matrix, which has the intuitive appeal of minimizing the volume of the confidence ellipsoid for the parameter estimates $\hat{\boldsymbol{\beta}}_{\boldsymbol{t}}^{\boldsymbol{f}}$ and $\hat{\boldsymbol{\beta}}_{\boldsymbol{t}}^{\boldsymbol{p}}$; refer to Raghavarao et al. (2010) for further discussion.

Unfortunately, there are no computationally efficient methods to identify the price vector $\boldsymbol{p}_{\boldsymbol{t}}^*$ that maximizes the determinant of the Fisher information matrix. Therefore, when pricing to learn, one can first try enumerating all possible price vectors in $\mathcal{P}_t$ and evaluating the determinant of the Fisher information matrix for each one, choosing the price vector with the largest determinant; for feasible price sets and assortments that are not too large, this approach provides an optimal solution in a reasonable amount of time. In fact, for the implementation of the *Fast Learning and Pricing for Varying Assortments* algorithm at Zenrez, we found that this approach was computationally feasible for over 95% of the assortments, enabling us to calculate and use the optimal price vector. If complete enumeration is computationally infeasible, then we suggest pursuing one or both of the following two heuristics.

For the first heuristic, one can employ the methods of swapping, cycling, and re-labeling to identify a solution; although optimality is not guaranteed, these heuristics are well-studied and commonly used in conjoint analysis and are detailed in Sandor and Wedel (2001). For the second heuristic, one can employ grid search by reducing the set of possible price vectors $\mathcal{P}_t$ to a computationally-feasible size, evaluating the determinant of the Fisher information matrix for each price vector in the reduced set, and choosing the one with the largest determinant. In our field experiment with Zenrez, we used grid search for the $< 5\%$ of assortments where complete enumeration was not computationally feasible.

During the *pricing to earn* phase, Algorithm 1 selects the price vector $\boldsymbol{p}_{\boldsymbol{t}}^*$ from the set of feasible price vectors $\mathcal{P}_t$ in a greedy fashion by assuming that the current parameter estimates $\hat{\boldsymbol{\beta}}_{\boldsymbol{t}}^{\boldsymbol{f}}$ and $\hat{\boldsymbol{\beta}}_{\boldsymbol{t}}^{\boldsymbol{p}}$ are the true parameters and maximizing current-period revenue under this assumption:

$$\boldsymbol{p}_{\boldsymbol{t}}^* = \arg\max_{\boldsymbol{p}_{\boldsymbol{t}} \in \mathcal{P}_t} \sum_{i=0}^{N_t} p_{it} \cdot \frac{\exp(\boldsymbol{x}_{\boldsymbol{it}}^\intercal \hat{\boldsymbol{\beta}}_{\boldsymbol{t}}^{\boldsymbol{f}} - \boldsymbol{x}_{\boldsymbol{it}}^\intercal \hat{\boldsymbol{\beta}}_{\boldsymbol{t}}^{\boldsymbol{p}} p_{it})}{\sum_{l=0}^{N_t} \exp(\boldsymbol{x}_{\boldsymbol{lt}}^\intercal \hat{\boldsymbol{\beta}}_{\boldsymbol{t}}^{\boldsymbol{f}} - \boldsymbol{x}_{\boldsymbol{lt}}^\intercal \hat{\boldsymbol{\beta}}_{\boldsymbol{t}}^{\boldsymbol{p}} p_{lt})} \,. \tag{5}$$

Given our assumption that demand is independent and identically distributed across all customers shopping assortment t, note that maximizing total revenue from assortment t is equivalent to maximizing the expected revenue from a single customer shopping assortment t. We may also face computational challenges when pricing to earn. Thus, in order to solve (5), we recommend the following approach. Similar to the *pricing to learn* phase, we first try enumerating all possible price vectors in $\mathcal{P}_t$ and evaluating the expected current-period revenue of each one, choosing the price vector with the largest expected revenue; for feasible price sets and assortments that are not too large, this approach can solve (5) to optimality in a reasonable amount of time. In fact, for the implementation of the *Fast Learning and Pricing for Varying Assortments* algorithm at Zenrez, we found that this approach was computationally feasible for over 95% of the assortments, enabling us to calculate and use the optimal price vector. If complete enumeration is computationally infeasible, then we suggest pursuing one or both of the following two approaches.

For the first approach, we recommend using the parametric linear programming methodology recently proposed in Sumida et al. (2021) to identify the optimal price vector; refer to Section 6.3 in Sumida et al. (2021) for a description of how to apply their methodology to our setting. For the second approach, one can employ grid search by reducing the set of possible price vectors $\mathcal{P}_t$ to a computationally-feasible size, evaluating the expected current-period revenue of each price vector in the reduced set, and choosing the one with the largest expected revenue. This is a common heuristic used in practice; see, e.g., Kannan et al. (2009), Iyengar et al. (2011), and Hosanagar et al. (2008). In our field experiment with Zenrez, we used grid search for the $< 5\%$ of assortments where complete enumeration was not computationally feasible.

Regardless of the pricing phase, at the end of each period our algorithm observes demand $\boldsymbol{y_t}$ and updates the parameter estimates $\hat{\lambda}$, $\hat{\boldsymbol{\beta}}^{\boldsymbol{f}}_{\boldsymbol{t+1}}$ and $\hat{\boldsymbol{\beta}}^{\boldsymbol{p}}_{\boldsymbol{t+1}}$ with their maximum likelihood estimators using all data observed through period t, $(\boldsymbol{p^*_s}, \boldsymbol{x_{is}} \forall i \in \{1, ..., N_s\}, \boldsymbol{y_s} \quad \forall s \leq t)$. Thus we note that even when pricing to earn, Algorithm 1 continues to passively learn and improve its parameter estimates. For retailers who do not observe the number of customer arrivals m_t in each period t (e.g., many brick-and-mortar retailers), we refer the reader to Newman et al. (2014) for an estimation routine that jointly estimates the arrival rate and parameters of the MNL model.

Finally, before advancing to the next period, our algorithm uses *SWITCH* - the Boolean switching criteria test specified as an input - to decide when to transition from the *pricing to learn* phase to the *pricing to earn* phase. A simple switching criteria would be to pre-select the switching period τ so that the pricing phase is *pricing to learn* for periods $t = 1, ..., \tau$ and *pricing to earn* for periods $t = \tau + 1, ..., T$. This approach is commonly used in other demand learning and pricing algorithms that follow a learn-then-earn approach, e.g., Besbes and Zeevi (2012). Alternatively, we note that any data observed through period t, i.e., $(\boldsymbol{p^*_s}, \boldsymbol{x_{is}} \forall i \in \{1, ..., N_s\}, \boldsymbol{y_s} \quad \forall s \leq t)$, could be used as an input to *SWITCH* in order to help dynamically choose when to switch from the *pricing to learn* phase to the *pricing to earn* phase.

There are a variety of practical considerations that can be incorporated into the switching criteria test. For example, retailers may choose to incorporate a notion of stability of parameter estimates: if the parameters estimated by the algorithm using additional data begin to converge - i.e., they do not change much given the current period's observed demand - then the value of learning may no longer outweigh the value of earning, signaling a good time to transition phases. Similarly, retailers may choose to incorporate a notion of stability of prices: if the prices recommended by the algorithm do not change when using recent periods' parameter estimates, there may not be much value in learning more precise parameter estimates. An additional consideration is the length of the time horizon T; a retailer facing a small time horizon T may want to switch phases earlier than a retailer facing a much longer horizon. We describe our choice for $SWITCH$ in our implementation at Zenrez in Section 4.1. Finally, we point out that knowledge of the length of the time horizon T is not required to run our algorithm, although it may be estimated and used as an input to $SWITCH$ if desired.

3.1. Comparison with M3P Algorithm

As described in Section 1.1, only one other paper (Javanmard et al. (2020)) presents a demand learning and pricing algorithm, M3P, for a (nearly) identical setting to ours. M3P alternates between learning and earning phases in an episodic manner, where in each learning phase it selects a random price vector for each customer, and in each earning phase it selects a price vector that maximizes revenue based on the parameter estimates derived from the learning phases. Episode k requires $k+l$ customers, where l is a constant number of customers shown random prices in the learning phase and k is a linearly increasing number of customers shown (expected) revenue-maximizing prices in the earning phase. In this section, we compare the performance of our *Fast Learning and Pricing for Varying Assortments* algorithm with the M3P algorithm for various values of l via simulations.

For our simulations, we consider a retailer selling three products per assortment, i.e., $N_t = 3\ \forall t = 1, ..., T$, plus an outside option. The set of possible prices $\mathcal{P}_{it}$ for each product $i = 1, 2, 3$ is $\{12, 13, 14, 15, 16, 17, 18\}$. Each of the three products is characterized by two features that take values in [0,1], one of which interacts with price in the utility function; we also allow for price to enter the utility function independently of product features. To implement this using our general notation in Section 2, we used $d = 3$ features, where $x_{1it} \in [0, 1]$, $x_{2it} \in [0, 1]$, and $x_{3it} = 1\ \forall i = 1, 2, 3$ and $t = 1, ..., T$, with $\beta_3^f = 0$ and $\beta_2^p = 0$.

We conducted 200 simulations, where for each simulation, we first independently drew utility model parameters $\beta_1^f \sim Uniform(10, 20)$, $\beta_2^f \sim Uniform(10, 20)$, $\beta_1^p \sim Uniform(0, 1)$, and $\beta_3^p \sim Uniform(0, 1)$. For each simulation and for each assortment $t = 1, ..., T$, we randomly selected feature $x_{dit} \sim Uniform(0, 1)$ for $d = 1, 2$ and $i = 1, 2, 3$. We assume that $m_t = 20$ customers arrive and make a purchase decision in each assortment $t = 1, ..., T$. Finally, for each simulation, each customer j's random component of utility, ϵ_{ijt}, is drawn independently from a standard Gumbel distribution for all $i = 0, ..., 3$, $j = 1, ..., m_t$, and $t = 1, ..., T$. Given the set of utility model parameters, features, and realized random components of utilities (i.e., $\boldsymbol{\beta^f}, \boldsymbol{\beta^p}, \boldsymbol{x_{it}}$, and ϵ_{ijt} $\forall i, j, t$) drawn independently for each of the 200 simulations, we implemented our *Fast Learning and Pricing for Varying Assortments* algorithm and the M3P algorithm with four different values for the number of customers in each learning phase - $l = 1, 5, 10$, and 15 - and compared their average performance across all 200 simulations.

We implemented our *Fast Learning and Pricing for Varying Assortments* algorithm as specified in Algorithm 1. For the switching criteria test ($SWITCH$) input to our algorithm, we used an identical test to what we used for our field experiment with Zenrez; please see Section 4.1 for details, noting that we generated four hypothetical classes instead of ten for Criterion 3. We implemented M3P as specified in Javanmard et al. (2020) with the only exception being that we required prices to be discrete, constrained to the same set of possible price vectors $\mathcal{P}_t$ that our algorithm faced. With this added constraint, the solution techniques proposed for the earning phases in Javanmard et al. (2020) were not applicable and we instead solved for the optimal prices in both the M3P and *Fast Learning and Pricing for Varying Assortments* algorithms by enumerating and evaluating all possible price vectors. It is also worth noting that one of the modeling differences between our work and Javanmard et al. (2020) is that they allow a different price vector to be offered to each customer shopping assortment t, whereas we require a single price vector $\boldsymbol{p_t}$ to be offered to all customers shopping assortment t. Thus when implementing M3P in our simulations, we allowed each customer to be offered a different price vector, even for customers shopping the same assortment.

As a benchmark for comparing the performance of each of the algorithms, we consider the (unrealistic) case where the retailer knows $\boldsymbol{\beta^f}$ and $\boldsymbol{\beta^p}$ at the start of the selling season and for each assortment t, uses this knowledge to offer the expected revenue-maximizing price vector:

$$\boldsymbol{p_t^*} = \arg\max_{\boldsymbol{p_t} \in \mathcal{P}_t} \sum_{i=0}^{3} p_{it} \cdot \frac{\exp(\boldsymbol{x_{it}^\intercal}\boldsymbol{\beta^f} - \boldsymbol{x_{it}^\intercal}\boldsymbol{\beta^p} p_{it})}{\sum_{l=0}^{3} \exp(\boldsymbol{x_{lt}^\intercal}\boldsymbol{\beta^f} - \boldsymbol{x_{lt}^\intercal}\boldsymbol{\beta^p} p_{lt})} . \tag{6}$$

Note that (6) is the objective of the *pricing to earn* phase in Algorithm 1, with parameter estimates $\boldsymbol{\hat{\beta}_t^f}$ and $\boldsymbol{\hat{\beta}_t^p}$ replaced with their true values $\boldsymbol{\beta^f}$ and $\boldsymbol{\beta^p}$. Denote $\hat{r}_{ts}^{OPT}$ as the revenue earned in period t and simulation s when following this benchmark pricing policy. Similarly, denote $\hat{r}_{ts}^{ALG}$ as the revenue earned in period t and simulation s when following one of the other implemented algorithms, ALG, where ALG can be either our *Fast Learning and Pricing for Varying Assortments* algorithm or M3P with $l = 1, 5, 10$, or 15. To compare each algorithm with the benchmark policy,

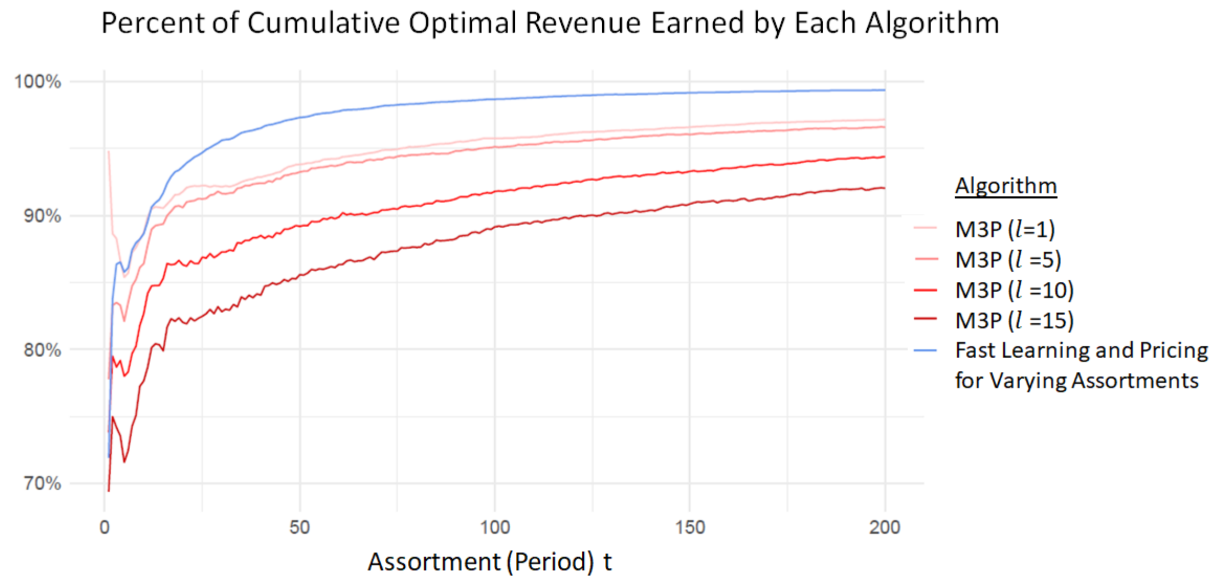

Figure 2 **Percent of cumulative optimal revenue earned by our Fast Learning and Pricing for Varying Assortments algorithm and M3P with $l = 1, 5, 10,$ and 15.**

we calculate the "percent of cumulative optimal revenue earned" for each period $t = 1, ..., T$ and for each ALG, as an average over all simulations:

$$\frac{1}{200}\sum_{s=1}^{200}\left(\frac{\sum_{l=1}^{t}\hat{r}_{ls}^{ALG}}{\sum_{l=1}^{t}\hat{r}_{ls}^{OPT}}\right). \tag{7}$$

Figure 2 plots the percent of cumulative optimal revenue earned by each of the algorithms implemented for each t=1,...,200. We note that the simulations were conducted for $T = 200$ periods, although since the value T is not used in any of the algorithm implementations, one can easily compare algorithm performance for any value of $T \leq 200$. Figure 2 shows that after the first few periods, our *Fast Learning and Pricing for Varying Assortments* algorithm outperforms all implementations of M3P by a considerable margin. Importantly, our algorithm is able to achieve these results even though it allows for only a single price vector $\boldsymbol{p}_t^*$ to be offered in each assortment t, whereas M3P offers multiple price vectors in a single assortment. Furthermore, by period 25, our algorithm has attained 95% of the cumulative optimal revenue earned over those periods, highlighting our algorithm's ability to learn utility model parameters and set (near-)optimal prices very quickly. For the same milestone, it takes 75 periods for M3P with $l = 1$ and 100 periods for M3P with $l = 5$; for larger l - equating to longer learning phases - it is clear that M3P spends too much time in the learning phases and is not able to converge to the optimal prices quickly.

We believe the key reason for these results lies in how the algorithms learn demand. Our algorithm conducts its *pricing to learn* phase in the initial periods and can then capitalize on that learning for the remainder of the season, whereas M3P spreads its learning episodically over the entire season. Furthermore, our algorithm learns by offering the most informative price vectors - i.e., those that maximize the determinant of the Fisher information matrix for $(\boldsymbol{\beta^f}, \boldsymbol{\beta^p})$ - whereas M3P learns by offering random price vectors.

4. Field Experiment

For the development and implementation of our work, we collaborated with Zenrez, an e-commerce company that partners with fitness studios across the United States and Canada to sell excess capacity of fitness studio classes. Every night at 9:00pm, Zenrez posts classes (i.e., products) that have remaining capacity for the following day and offers them at a discounted price via a widget located on the partner studios' webpages or app. When a user views the widget, they see all next-day classes offered by Zenrez for that fitness studio (see, e.g., Figure 1). Each class is characterized by features such as class type (e.g., yoga, spin, etc.), duration, and day of week, and data supports that it is reasonable to assume that a vast majority of consumers choose at most one class from each daily assortment.

The assortment of classes changes each day, and prices can vary across assortments but are fixed within each assortment in order to avoid negative customer perception (e.g., once classes are posted, their prices do not change). Zenrez has the flexibility to choose an integer price for each class within a studio-specified interval, [floor, ceiling], and earns a commission proportional to the selling price for each class sold via their widget; note that the marginal cost to Zenrez of selling an additional unit of capacity is negligible and thus revenue and profit are equivalent. The appeal of purchasing last-minute classes from Zenrez rather than directly from the studio is that Zenrez sells the classes at a discount. Since studios offer various types of volume discounts for frequent customers, Zenrez's customers are predominantly infrequent customers, supporting our assumption that demand is independent over time.

4.1. Algorithm Implementation

To implement our *Fast Learning and Pricing for Varying Assortments* algorithm, we developed a fully-automated pricing tool at Zenrez. Since assortments change every day, the length of each period t is set to one day. The tool is run automatically every day for a given studio, setting prices for the following day's assortment of classes. Given the diversity across studios in terms of their classes offered, locations, etc., as well as the observation that most customers only view classes on Zenrez's widgets from a single studio on a given day, we ran Algorithm 1 separately for each studio. The average number of classes offered each day (N_t) varied by studio, ranging from 5 to 15. Features used in our demand model included: class type indicators, class duration, day of week indicators, time of day indicators, and a star-instructor indicator.

To solve for the optimal price vectors in the *pricing to learn* and *pricing to earn* phases, we first checked if the number of possible price vectors in $\mathcal{P}_t$ was less than a studio-specific threshold that we considered computationally feasible for complete enumeration and evaluation to optimality. This threshold was determined based on extensive simulations using Zenrez's historical data, as well as the permissible run-time per studio (i.e., the time that prices must be set minus the time that the assortment of classes is finalized by the studio). If the number of possible price vectors was less than the threshold, then we enumerated all possible price vectors, and for each one, evaluated either the determinant of the Fisher information matrix for the *pricing to learn* phase or the expected current-period revenue for the *pricing to earn* phase; finally, we chose the price vector which gave the maximum value. This was the approach used for over 95% of the assortments in our field experiment, enabling us to calculate and use the optimal price vector. For the other $<5\%$ of the assortments in which the number of possible price vectors in $\mathcal{P}_t$ surpassed the threshold, we used grid search to evaluate only a subset of the price vectors in $\mathcal{P}_t$, and chose the one with either the largest determinant of the Fisher information matrix in the *pricing to learn* phase or the largest expected revenue in the *pricing to earn* phase. To choose the subset of prices, for each class i, we selected prices from feasible price set $\mathcal{P}_{it}$ that were (approximately) equally spaced between and including $\min\{\mathcal{P}_{it}\}$ and $\max\{\mathcal{P}_{it}\}$.

For the switching criteria test ($SWITCH$) input to our algorithm, we used a test of price stability: intuitively, if the revenue-maximizing prices do not change when using recent periods' parameter estimates, there is likely little value in learning more precise parameter estimates, and our algorithm switches from the *pricing to learn* phase to the *pricing to earn* phase. Specifically, $SWITCH$ evaluates to $TRUE$ (and therefore triggers our algorithm to switch from the *pricing to learn* phase to the *pricing to earn* phase) when the following three criteria are satisfied:

1. Minimum time in *pricing to learn* phase: $t \geq 3$.
2. Price stability for current assortment:

$$\begin{aligned}
&\arg\max_{\boldsymbol{p_t} \in \mathcal{P}_t} \sum_{i=0}^{N_t} p_{it} \cdot \frac{\exp(\boldsymbol{x}_{it}^\intercal \hat{\boldsymbol{\beta}}_{\boldsymbol{t}}^{\boldsymbol{f}} - \boldsymbol{x}_{it}^\intercal \hat{\boldsymbol{\beta}}_{\boldsymbol{t}}^{\boldsymbol{p}} p_{it})}{\sum_{l=0}^{N_t} \exp(\boldsymbol{x}_{lt}^\intercal \hat{\boldsymbol{\beta}}_{\boldsymbol{t}}^{\boldsymbol{f}} - \boldsymbol{x}_{lt}^\intercal \hat{\boldsymbol{\beta}}_{\boldsymbol{t}}^{\boldsymbol{p}} p_{lt})} = \arg\max_{\boldsymbol{p_t} \in \mathcal{P}_t} \sum_{i=0}^{N_t} p_{it} \cdot \frac{\exp(\boldsymbol{x}_{it}^\intercal \hat{\boldsymbol{\beta}}_{\boldsymbol{t-1}}^{\boldsymbol{f}} - \boldsymbol{x}_{it}^\intercal \hat{\boldsymbol{\beta}}_{\boldsymbol{t-1}}^{\boldsymbol{p}} p_{it})}{\sum_{l=0}^{N_t} \exp(\boldsymbol{x}_{lt}^\intercal \hat{\boldsymbol{\beta}}_{\boldsymbol{t-1}}^{\boldsymbol{f}} - \boldsymbol{x}_{lt}^\intercal \hat{\boldsymbol{\beta}}_{\boldsymbol{t-1}}^{\boldsymbol{p}} p_{lt})} \\
&= \arg\max_{\boldsymbol{p_t} \in \mathcal{P}_t} \sum_{i=0}^{N_t} p_{it} \cdot \frac{\exp(\boldsymbol{x}_{it}^\intercal \hat{\boldsymbol{\beta}}_{\boldsymbol{t-2}}^{\boldsymbol{f}} - \boldsymbol{x}_{it}^\intercal \hat{\boldsymbol{\beta}}_{\boldsymbol{t-2}}^{\boldsymbol{p}} p_{it})}{\sum_{l=0}^{N_t} \exp(\boldsymbol{x}_{lt}^\intercal \hat{\boldsymbol{\beta}}_{\boldsymbol{t-2}}^{\boldsymbol{f}} - \boldsymbol{x}_{lt}^\intercal \hat{\boldsymbol{\beta}}_{\boldsymbol{t-2}}^{\boldsymbol{p}} p_{lt})}.
\end{aligned} \tag{8}$$

3. Price stability for random assortment: First, generate a random assortment of $i = 1, ..., 10$ hypothetical classes, where class i has feature vector $\boldsymbol{x_i} \in \mathbb{R}^d$ randomly selected (with equal probability) from the set of all possible attribute values for each feature offered by that studio in its history. Set the feasible price vector $\mathcal{P}$ for each price $\boldsymbol{p} = \{p_1, ..., p_{10}\}$ to be all integers

between and inclusive of the minimum of all floor prices and maximum of all ceiling prices offered by that studio in its history. Finally, we check the following equalities for price stability:

$$\arg\max_{\boldsymbol{p}\in\mathcal{P}} \sum_{i=0}^{10} p_i \cdot \frac{\exp(\boldsymbol{x_i}^\intercal \hat{\boldsymbol{\beta}}_{\boldsymbol{t}}^{\boldsymbol{f}} - \boldsymbol{x_i}^\intercal \hat{\boldsymbol{\beta}}_{\boldsymbol{t}}^{\boldsymbol{p}} p_i)}{\sum_{l=0}^{10} \exp(\boldsymbol{x_l}^\intercal \hat{\boldsymbol{\beta}}_{\boldsymbol{t}}^{\boldsymbol{f}} - \boldsymbol{x_l}^\intercal \hat{\boldsymbol{\beta}}_{\boldsymbol{t}}^{\boldsymbol{p}} p_l)} = \arg\max_{\boldsymbol{p}\in\mathcal{P}} \sum_{i=0}^{10} p_i \cdot \frac{\exp(\boldsymbol{x_i}^\intercal \hat{\boldsymbol{\beta}}_{\boldsymbol{t-1}}^{\boldsymbol{f}} - \boldsymbol{x_i}^\intercal \hat{\boldsymbol{\beta}}_{\boldsymbol{t-1}}^{\boldsymbol{p}} p_i)}{\sum_{l=0}^{10} \exp(\boldsymbol{x_l}^\intercal \hat{\boldsymbol{\beta}}_{\boldsymbol{t-1}}^{\boldsymbol{f}} - \boldsymbol{x_l}^\intercal \hat{\boldsymbol{\beta}}_{\boldsymbol{t-1}}^{\boldsymbol{p}} p_l)}$$
$$= \arg\max_{\boldsymbol{p}\in\mathcal{P}} \sum_{i=0}^{10} p_i \cdot \frac{\exp(\boldsymbol{x_i}^\intercal \hat{\boldsymbol{\beta}}_{\boldsymbol{t-2}}^{\boldsymbol{f}} - \boldsymbol{x_i}^\intercal \hat{\boldsymbol{\beta}}_{\boldsymbol{t-2}}^{\boldsymbol{p}} p_i)}{\sum_{l=0}^{10} \exp(\boldsymbol{x_l}^\intercal \hat{\boldsymbol{\beta}}_{\boldsymbol{t-2}}^{\boldsymbol{f}} - \boldsymbol{x_l}^\intercal \hat{\boldsymbol{\beta}}_{\boldsymbol{t-2}}^{\boldsymbol{p}} p_l)}. \tag{9}$$

Criterion 1 requires that we stay in the *pricing to learn* phase for at least three periods, essentially to provide different parameter estimates for Criteria 2 and 3. Criterion 2 requires that the revenue-maximizing price vectors from (5) are identical for the current period's assortment when using the last three periods' parameter estimates. Similarly, for robustness, Criterion 3 requires that the revenue-maximizing price vectors are identical for a random assortment when using the last three periods' parameter estimates. Meeting Criteria 2 and 3 signals that the parameter estimates are likely sufficiently stable.

We chose these criteria for *SWITCH* after testing many other possibilities in our simulations presented in Section 3.1. For example, and in line with many academic papers (e.g., Besbes and Zeevi (2012)), one possibility of *SWITCH* that we tested was pre-selecting a switching period τ, independent of any studio-specific parameters. Another possibility of *SWITCH* we tested included only the first two criteria described above, omitting Criterion 3; this tended to result in the algorithm prematurely switching from the *pricing to learn* to *pricing to earn* phase, particularly when a studio only offered a few classes on a given day. We also evaluated tests of parameter stability, for example by requiring the difference between parameter estimates from the last three periods to be less than a threshold; this tended not to work as well as our price stability *SWITCH* test because for some parameters, although they still experienced changes larger than the threshold, these changes did not impact the revenue-maximizing prices and therefore there was little value in learning these parameters with more precision.

4.2. Experimental Design

To evaluate the effectiveness of our *Fast Learning and Pricing for Varying Assortments* algorithm, we conducted a three-month, controlled field experiment where prices for studios in the treatment group were set according to our algorithm while prices for studios in the control group were set according to Zenrez's existing pricing practices. For years, and like many retailers, Zenrez had been pricing classes proportional to the historical utilization (percent of capacity sold) of near-identical classes – those with the same instructor, class type, time of day, day of week, and duration (if such classes existed). The most popular classes were priced at or near the ceiling of the feasible price set, and the least popular classes were priced at or near the floor of the feasible price set. If near-identical classes did not exist for a class - which happened often - Zenrez would typically price the class in the middle of the feasible price set. Our objective was to identify the impact of our *Fast Learning and Pricing for Varying Assortments* algorithm on Zenrez's primary metric of interest: average daily revenue. Thus, the treatment effect we measured was the percent increase in average daily revenue between studios in the treatment group compared to the control group for each of the three months in the experiment. It is worth noting that although the field experiment was pre-specified to last for three months, this length of time is independent of the length of the selling season T that a given studio may face; instead, T is determined by things such as management's planning horizon, changes in the competitive landscape, and fitness trends. Thus, for selling seasons lasting longer than 3 months - as is likely the case for many of the studios - we would expect that results achieved in the *pricing to earn* phase of our algorithm in month 3 would persist until the end of the season.

We identified a set $\mathcal{S}$ of 52 studios whose average daily revenue exceeded a minimum threshold over the six months prior to the field experiment (the "pre-period") and who had been continuously operating the widget during this pre-period. Across these 52 studios, there was significant heterogeneity in pre-period revenue and trends, which would make a simple difference-in-means or difference-in-differences evaluation of treatment effects unreliable. Therefore, we decided to use synthetic controls to estimate treatment effects. The synthetic controls method combines elements of matching and difference-in-differences techniques to account for heterogeneity in pre-period trends by finding a weighted average of studios in the control group whose trend in the pre-period closely matches the pre-period trend for the studios in the treatment group; the trend that we matched on was average daily revenue during each of the six months in the pre-period.

Specifically, define $\mathcal{T}$ to be the set of studios in the treatment group, where prices were set using Algorithm 1, and define $\mathcal{C} = \mathcal{S} \setminus \mathcal{T}$ to be the set of studios in the control group, where prices were set using Zenrez's existing pricing practices. Let r_{sm} be the average daily revenue of studio $s \in \mathcal{S}$ during month m. Considering pre-period months $m = 1, ..., 6$, the synthetic control, $w_s^* \ \forall s \in \mathcal{C}$, is calculated as

$$\begin{aligned} \arg\min \sum_{m=1}^{6} &\left(\frac{1}{|\mathcal{T}|} \sum_{s \in \mathcal{T}} r_{sm} - \sum_{s \in \mathcal{C}} w_s r_{sm} \right)^2 \\ \text{s.t.} &\sum_{s \in \mathcal{C}} w_s = 1 \\ &w_s \geq 0 \ \forall s \in \mathcal{C} \end{aligned} . \tag{10}$$

Intuitively, w_s^* can be interpreted as the optimal weights assigned to control studios $s \in \mathcal{C}$ which provide a weighted average of control studios that best represents the average treatment studio with respect to pre-period average daily revenue. If a high-quality synthetic control is obtained, then any difference in average daily revenue during the three-month experiment between the average treatment studio and its synthetic control can be attributed to the impact of our algorithm. Refer to Abadie et al. (2010) for a detailed description of the synthetic controls method.

To assign the 52 studios into treatment and control groups, it was important to ensure that we would have a high-quality pre-period match between weighted average daily revenue of studios in the control group and average daily revenue of studios in the treatment group, which would allow us to effectively evaluate the outcome of the experiment. To do so, we iteratively conducted the following steps, initially starting with all studios being unassigned to the treatment or control group:

(i) For each unassigned studio, find its synthetic control using (10), where $\mathcal{T}$ is the unassigned studio and $\mathcal{C}$ contains all of the other unassigned studios and studios already assigned to the control group.

(ii) Choose the studio that has the best synthetic control, i.e., the studio that achieves the smallest objective value from (10).

(iii) Assign this studio to the treatment group.

(iv) Assign studio s included in its synthetic control to the control group (if not already there) as long as $w_s^* \geq 0.02$. Without applying a small threshold, too many studios are assigned to the control group which do not significantly impact the quality of the synthetic control, and only a few treatment studios would be identified.

We repeated these steps until all studios were assigned to the treatment or control groups. Using this iterative approach, each of the last three unassigned studios had a poor-quality synthetic control from the set of other unassigned studios and the control studios, so we assigned these final three studios to the control group. Our approach resulted in $|\mathcal{T}| = 23$ studios in the treatment group and $|\mathcal{C}| = 29$ studios in the control group. It is worth noting that studios did not know that they were included in the experiment.

Given our approach used to identify studios for the treatment and control groups, our synthetic control, $w_s^*\ \forall s \in \mathcal{C}$, was a near-perfect match to the average of studios in the treatment group with respect to our matching criteria - average daily revenue. Because of this, we can attribute the difference in average daily revenue during the three-month experiment to the impact of our algorithm. Furthermore, Table 1 shows that our approach resulted in balance across studios in the treatment and control groups over the six-month pre-period on a variety of characteristics. Note that only 1.7% of the classes offered by Zenrez sell out during the pre-period, and thus we assume that all demand can be met.

	Treatment	Synthetic Control
Avg. Daily Revenue	1.00	1.00
Avg. Daily Purchases	1.00	0.95
Avg. Daily Classes	1.00	0.95
Avg. Price per Class	1.00	1.06
Avg. Widget Traffic	1.00	1.08
Unique Cities	11	13
Sell-Out Percent	1.6%	1.7%

Table 1 Summary statistics for the studios in the treatment and control groups over the six-month pre-period. All metrics except Unique Cities and Sell-Out Percent are normalized to the studios in the treatment group's average pre-period values (averaged across all studios in the treatment group and all six pre-period months).

4.3. Results

The treatment effect that we measured was the percent increase in average daily revenue for each of the three months in the field experiment, comparing average daily revenue across all studios in the treatment group with its synthetic control. Specifically, our revenue treatment effect, R_m^*, is calculated for each month m as

$$R_m^* = \frac{\frac{1}{|\mathcal{T}|}\sum_{s\in\mathcal{T}} r_{sm} - \sum_{s\in\mathcal{C}} w_s^* r_{sm}}{\sum_{s\in\mathcal{C}} w_s^* r_{sm}} \ . \tag{11}$$

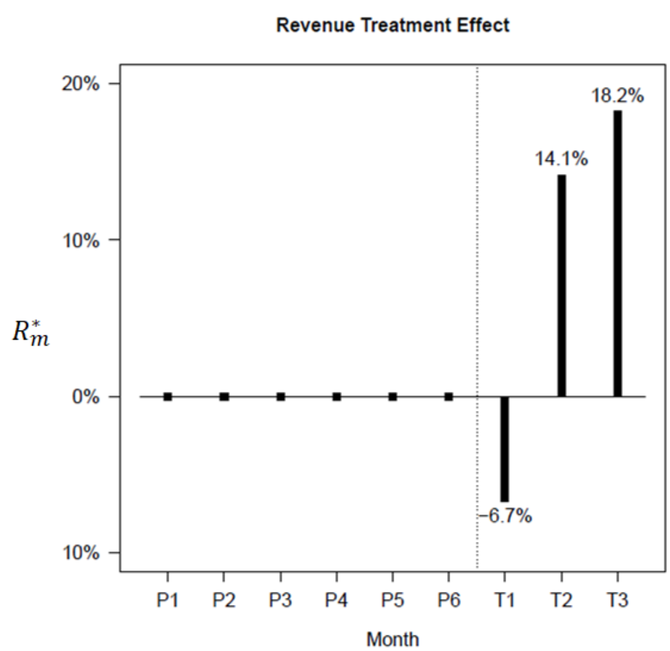

Figure 3 Percent increase in average daily revenue of studios in the treatment group vs. that of their synthetic control over the six-month pre-period (P1-P6) and three-month experiment (T1-T3). The start of the experiment is indicated by a vertical dashed line.

Figure 3 illustrates the results of our experiment. First, we can see that we were able to attain a very strong synthetic control; there is a near perfect match in average daily revenue for each of the six pre-period months (P1-P6) between treatment and synthetic control groups. Because of this, we attribute the difference between average daily revenue during the three-month experiment (T1-T3) to the impact of our algorithm. Compared to the synthetic control, the studios in the treatment group experienced a dip in average daily revenue of 6.7% in the first month of our experiment and an increase in average daily revenue of 14.1% and 18.2% in the second and third months of our experiment, respectively. We note that this dip followed by gain is consistent with our expectations that initial *pricing to learn* would yield a dip in revenue while subsequent *pricing to earn* would lead to higher revenue in the long run. Over the three-month experiment, studios in the treatment group experienced an 8.5% increase in average daily revenue compared to the synthetic control. Since the revenue gains were persistent at well above 10% across the second and third months, it is reasonable to expect that gains of a similar magnitude would endure over future periods if the algorithm were run for longer.

To measure the significance of our monthly revenue treatment effects, we performed Randomization Inference with Fisher's Exact Test to quantify the probability of observing our monthly treatment effects under the null hypotheses that our algorithm had no effect on revenue each month. Please see Appendix B for a description of the intuition behind the test and how we performed the test to evaluate our field experiment. For each month, we use Fisher's Exact Test to empirically calculate a one-sided p-value representing the probability that we would observe more than an R^*_m increase in revenue in month m due to chance. The p-values for months $m = 1, 2, 3$ are 0.769, 0.062, and 0.054, respectively. Thus, we conclude that the initial dip in revenue in month 1 is unsurprising under the null hypothesis, whereas we have sufficient evidence to reject the null hypotheses in months 2 and 3 at the 10% significance level and conclude that our algorithm had a strong positive effect on revenue.

The majority of studios switched from the *pricing to learn* to *pricing to earn* phase within 20 days of the algorithm's launch, while two studios took just over 30 days to make the switch. Importantly, we see that our algorithm only required a very short *pricing to learn* phase, and was quickly able to capitalize on that learning in the *pricing to earn* phase. This is a promising result for many retailers who may not want to or may not be able to change prices multiple times within each assortment: our results show that minimal price experimentation timed at the beginning of each new assortment can reap huge payoffs as the season progresses, owed to the efficient learning that our algorithm provides. Furthermore, when external factors change that may influence demand model parameters - such as a competitor studio opening across the street - fast learning allows retailers to restart the algorithm and quickly learn the new demand model parameters.

To better understand the effect of our *Fast Learning and Pricing for Varying Assortments* algorithm, we analyzed prices and sales for the studios in the treatment group during the three-month experiment; results are shown in Figure 4. From Figure 4(a), we can see that the average price of an offered class decreased by approximately 3.0% during the experiment compared to the six-month pre-period. Unsurprisingly, demand was larger for the less expensive classes, and thus we see a decrease by approximately 3.6% in the average selling price during the experiment. From Figure 4(b), we can see that the decrease of 3.6% in average selling price resulted in an increase of approximately 15.6% in units sold over the three-month experiment, and an increase of approximately 26.0% in the second and third months of the experiment when the algorithm was primarily pricing to earn. This substantial increase in quantity sold from only a slight decrease in average selling price led to the overall positive revenue results. It is worth noting that this increase in quantity sold only resulted in a slight increase in the percent of classes that sold out, from 1.6% in the pre-period to 1.9% during the experiment. Finally, Figure 4(c) shows that our algorithm increased daily price variance, especially in the first month of the experiment when our algorithm was predominately pricing to learn. This is because the maximally informative price set - i.e., the price set that maximizes the determinant of the Fisher information matrix - often prices some classes at their upper bound and others at their lower bound, leading to high price variance. This observation is consistent with the literature on conjoint analysis, which shows that numeric features are often set at or near their boundaries (Kanninen 2002).

4.4. Alternative Evaluation Approach

When applying synthetic controls to multiple treatment units (studios), there are two approaches that are commonly used in the literature. The approach that we used in Sections 4.2 and 4.3 averages (unweighted) all of the treatment units together, generates a single synthetic control for the average treatment unit, and then estimates the treatment effect on that single treatment unit (see, e.g., Kreif et al. (2016) and Robbins et al. (2017)). The second approach that can be used to apply the synthetic control method to multiple treatment units generates a synthetic control for each treatment unit separately, calculates the treatment effect for each treatment unit, and then averages the treatment effects across all treatment units (see, e.g., Dube and Zipperer (2015) and Abadie (2021)). In this section, we evaluate our field experiment using this second approach to provide additional evidence and insights regarding how Algorithm 1 performs in practice.

For each treatment studio $t \in \mathcal{T}$, define synthetic control w^*_{st} $\forall s \in \mathcal{C}$ as

$$\begin{aligned} \arg\min \sum_{m=1}^{6} (r_{tm} - \sum_{s \in \mathcal{C}} w_{st} r_{sm})^2 \\ \text{s.t.} \sum_{s \in \mathcal{C}} w_{st} = 1 \\ w_{st} \geq 0 \ \forall s \in \mathcal{C} \end{aligned} \quad . \tag{12}$$

Note that (12) is equivalent to (10) when $\mathcal{T}$ contains only a single treatment unit.

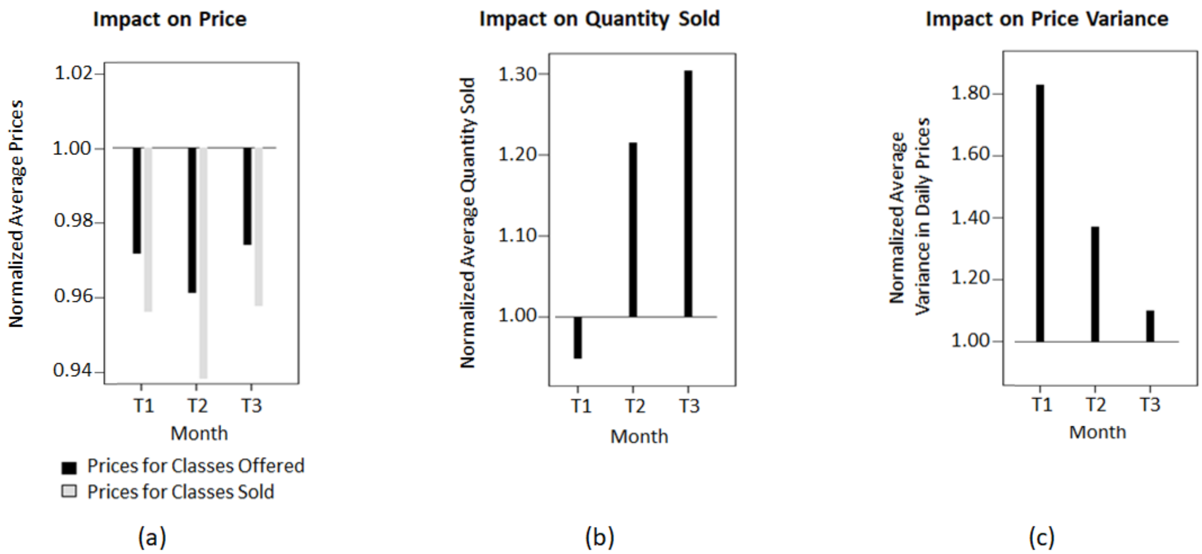

Figure 4 **Impact of Algorithm 1 on price, quantity sold, and price variance for each of the three months in our experiment, T1-T3. (a) Average price of classes offered and average price of classes sold by studios in the treatment group, normalized by the average price of classes offered by studios in the treatment group over the six-month pre-period. (b) Average quantity sold by studios in the treatment group, normalized by the average quantity sold by studios in the treatment group over the six-month pre-period. (c) Average variance in daily prices for studios in the treatment group, normalized by the average variance in daily prices for studios in the treatment group over the six-month pre-period.**

Using the synthetic controls w^*_{st}, we define the revenue treatment effect averaged over all studios $t \in \mathcal{T}$, R^{Mult}_m, for each month m as

$$R^{Mult}_m = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \left(\frac{r_{tm} - \sum_{s \in \mathcal{C}} w^*_{st} r_{sm}}{\sum_{s \in \mathcal{C}} w^*_{st} r_{sm}} \right) . \tag{13}$$

Figure 5 illustrates the results and includes intervals around each value of R^{Mult}_m which depict the interquartile ranges for the percent increase in average daily revenue across all treatment studios. Unsurprisingly, the quality of the synthetic controls for each treatment studio are not as strong as the quality of the synthetic control for the average treatment studio; refer to Appendix C for a detailed discussion. Nonetheless, we believe that the results of this approach still provide valuable insights and support our findings in Section 4.3. Specifically, studios in the treatment group experience a dip in average daily revenue of 10.0% in the first month of our experiment, followed by an increase in average daily revenue of 14.9% and 13.4% during the second and third months of our experiment, respectively. The magnitudes of these effects are in line with our findings in Section 4.3, although we have more trust in the results in Section 4.3 due to the higher-quality synthetic control. We cannot report on p-values for each studio in the treatment group because

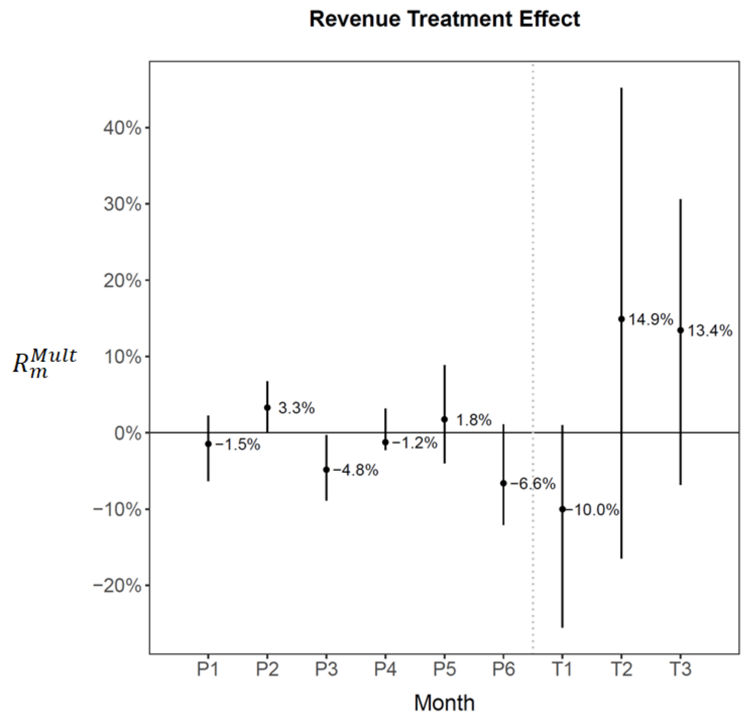

Figure 5 **Mean percent increase in average daily revenue of treatment studios vs. their synthetic controls, R_m^{Mult}, over the six-month pre-period (P1-P6) and three-month experiment (T1-T3). The intervals around each value of R_m^{Mult} depict the interquartile range for the percent increase in average daily revenue across all treatment studios. The start of the experiment is indicated by a vertical dashed line.**

the efficacy of Randomization Inference with Fisher's Exact Test critically depends on the quality of the synthetic controls.

Next, we use the synthetic controls w^*_{st} to estimate revenue treatment effects for the *pricing to learn* and *pricing to earn* phases - as opposed to monthly revenue treatment effects - which sheds more light on the impact of our algorithm over time. With slight abuse of notation, we define r_{sL} and r_{sE} to be the average daily revenue of studio s during that studio's *pricing to learn* and *pricing to earn* phases, respectively. We define the revenue treatment effect averaged over all studios $t \in \mathcal{T}$, R_m^{MultLE}, for each of the six pre-period months $m \in \{P1, P2, P3, P4, P5, P6\}$ and *pricing to learn* and *pricing to earn* phases $m \in \{L, E\}$, respectively, as

$$R_m^{MultLE} = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \left(\frac{r_{tm} - \sum_{s \in \mathcal{C}} w^*_{st} r_{sm}}{\sum_{s \in \mathcal{C}} w^*_{st} r_{sm}} \right) . \tag{14}$$

Figure 6(a) illustrates the results; note that the values of R_m^{MultLE} for each of the six pre-period months $m \in \{P1, P2, P3, P4, P5, P6\}$ are identical to those in Figure 5 since the same synthetic controls, w^*_{st}, are used. We see that the impact of our algorithm during the *pricing to learn* and *pricing to earn* phases ($m \in \{L, E\}$) supports our previous findings. Specifically, studios in the

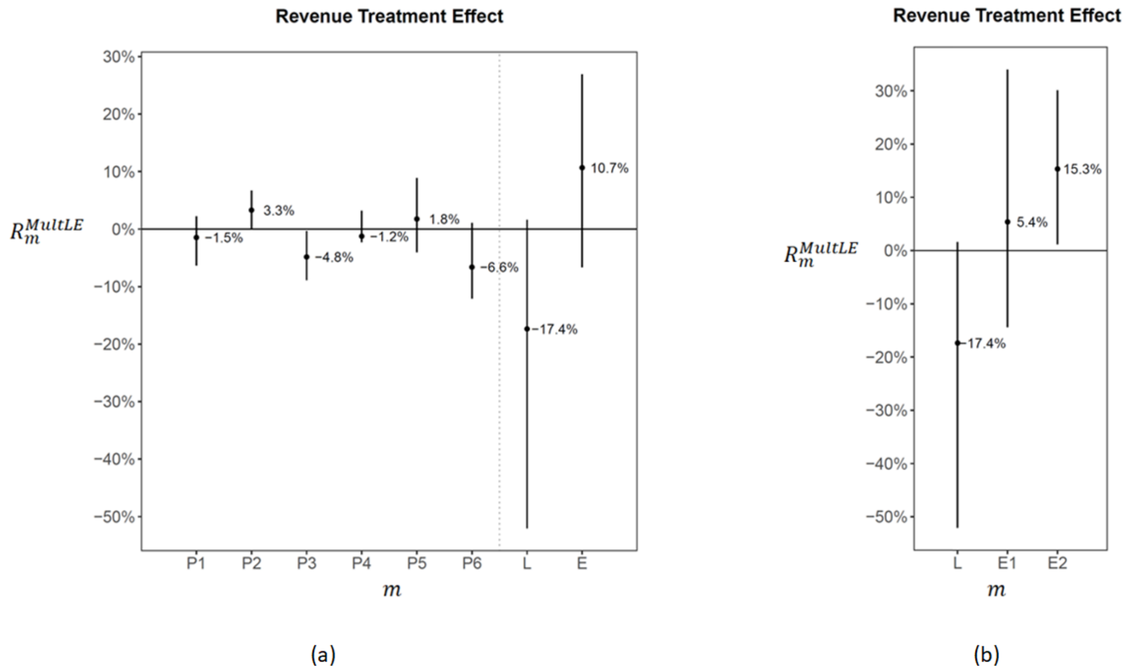

Figure 6 **Mean percent increase in average daily revenue of treatment studios vs. their synthetic controls, R_m^{MultLE}. The intervals around each value of R_m^{MultLE} depict the interquartile range for the percent increase in average daily revenue across all treatment studios. (a) R_m^{MultLE} over the six-month pre-period (P1-P6), pricing to learn phase (L), and pricing to earn phase (E); the start of the experiment is indicated by a vertical dashed line. (b) R_m^{MultLE} over the pricing to learn phase (L), first half of pricing to earn phase (E1), and second half of pricing to earn phase (E2).**

treatment group experience a substantial dip in average daily revenue of 17.4% during the short, initial *pricing to learn* phase, followed by a substantial increase in average daily revenue of 10.7% during the longer *pricing to earn* phase. Compared to the results in Figure 3, the 17.4% dip in average daily revenue in the *pricing to learn* phase is more than the 6.7% dip in average daily revenue of month 1 in the treatment period; this is to be expected since month 1 in the treatment period includes both *pricing to learn* and *pricing to earn* phases for most studios. The effect in the *pricing to earn* phase is perhaps a bit more surprising at first glance: compared to the results in Figure 3 which show an increase in average daily revenue of 14.1% and 18.2% in months 2 and 3 of the treatment period, respectively, the increase in average daily revenue in our *pricing to earn* phase is only 10.7%. It turns out that the reason behind this is that our algorithm still continues to passively learn throughout the *pricing to earn* phase, updating parameter estimates with their maximum likelihood estimator using all data observed up through the previous day. Thus, at the beginning of the final month of the experiment, our algorithm has already learned from both the *pricing to learn* phase and the beginning of the *pricing to earn* phase (typically more than half of this phase); in contrast, at the beginning of the *pricing to earn* phase, our algorithm has learned from only the *pricing to learn* phase.

To illustrate this point, we ran a very similar analysis to what was presented above, except instead of considering a single *pricing to earn* phase, we split the *pricing to earn* phase into two phases of equal length for each studio: *pricing to earn 1* ($E1$) and *pricing to earn 2* ($E2$). Figure 6(b) illustrates the results, where period E from Figure 6(a) has been divided into $E1$ and $E2$. These results indicate that indeed there is passive learning persisting beyond the initial *pricing to learn* phase. Specifically, in the first half of the *pricing to earn* phase, studios achieve an average increase in average daily revenue of 5.4%, while in the second half of the *pricing to earn* phase, studios achieve an average increase in average daily revenue of 15.3%; it is reasonable to expect that gains of a similar magnitude would endure over future periods if the algorithm were run for longer. A similar increase in the revenue treatment effect can be seen between months 2 and 3 in Figure 3, although the increase is less pronounced, perhaps due to the choice of months rather than *pricing to learn*, *pricing to earn 1*, and *pricing to earn 2* phases and the fact that studios in the treatment group have different lengths of each phase.

5. Conclusion

In this paper, we considered demand learning and pricing for a prevalent retail setting that has received little attention in the literature to date - namely, that of retailers who offer assortments of substitutable products that change frequently, e.g., due to limited inventory, perishable or time-sensitive products, or simply the retailer's desire to frequently offer new styles. Motivated by many retailers' desires to limit the number of price changes per assortment, we allow price changes to occur only when assortments change. We introduced a novel algorithm - *Fast Learning and Pricing for Varying Assortments* - that learns quickly in such an environment with varying assortments and limited price changes by adapting the commonly used marketing technique of conjoint analysis to our setting. A short learning phase is particularly important in our setting where retailers want to minimize negative customer perception from volatile price experimentation, as well as for retailers who have limited sales volume or a short selling season and thus cannot afford lengthy experimentation. Furthermore, when external factors change that may influence demand, a short learning phase allows retailers to restart the algorithm and quickly learn the new demand model parameters. Despite its importance in practice, the algorithmic characteristic of a short learning phase is rarely considered in the demand learning and pricing literature, and conjoint analysis has yet to be proposed for this task.

Importantly, and unlike most other demand learning and pricing papers, we implement our algorithm in practice and evaluate its effectiveness via a controlled field experiment. We illustrate how to use synthetic controls to estimate the treatment effect, a popular method recently proposed to evaluate policy interventions. Relative to the control group, our algorithm led to an expected initial dip in revenue during the *pricing to learn* phase, followed by a sustained and significant increase in average daily revenue of 14-18% throughout the *pricing to earn* phase. The results of our field experiment illustrate that our algorithmic contributions can make a significant impact in practice, and we hope that our work will inspire other retailers to implement our algorithm.

For future academic work, we believe that some of the same components of our algorithm could be used to help develop algorithms for other, related problems. For example, a natural extension of our model is to consider demand learning with joint price and assortment optimization, applicable for retailers who make these decisions simultaneously. Although we use conjoint analysis to help determine a single attribute value (price), conjoint analysis has historically been used to simultaneously determine multiple attribute values using a choice set that is constrained in size; thus we believe extending conjoint analysis for use in the learning phase of a joint price and assortment optimization problem might be possible.

More broadly, we hope that our work encourages other researchers to consider operations and marketing problems targeted for retailers with frequent assortment changes, limited sales volume, and/or an interest in limited price changes. Finally, we hope that our work serves as a motivating example to illustrate the benefit of borrowing ideas from multiple literatures - operations management, marketing, and computer science, in our case - to make a big impact in practice.

Acknowledgments

We sincerely thank Zenrez co-founders Matt Capizzi and Arthur Hong, as well as the entire Zenrez team, for their continuing support, sharing valuable business expertise through numerous discussions, and providing us with a considerable amount of time and resources to ensure a successful project. This research also benefitted from discussions during and after numerous seminars and with our colleagues; we thank all of the participants and appreciate their feedback. Finally, we thank the entire review team, whose comments significantly helped the presentation and analysis in this paper, and would like to specifically thank one of the anonymous reviewers for providing very detailed suggestions of alternative approaches and literature related to solving for the optimal price vector in the *pricing to earn* phase.

References

Abadie, Alberto. 2021. Using synthetic controls: Feasibility, data requirements, and methodological aspects. *Journal of Economic Literature* **59**(2) 391–425.

Abadie, Alberto, Alexis Diamond, Jens Hainmueller. 2010. Synthetic Control Methods for Comparative Case Studies: Estimating the Effect of California's Tobacco Control Program. *Journal of the American Statistical Association* **105**(490) 493–505.

Ban, Gah-Yi, Bora Keskin. 2020. Personalized dynamic pricing with machine learning: High dimensional features and heterogeneous elasticity. *Management Science* Forthcoming.

Besbes, Omar, Adam Elmachtoub, Yunjie Sun. 2020. Pricing analytics for rotable spare parts. *INFORMS Journal on Applied Analytics* **50**(5) 313–324.

Besbes, Omar, Assaf Zeevi. 2012. Blind Network Revenue Management. *Operations Research* **60**(6) 1537–1550.

Caro, Felipe, Jeremie Gallien. 2012. Clearance pricing optimization for a fast-fashion retailer. *Operations Research* **60**(6) 1404–1422.

Cavallo, Alberto. 2018. More amazon effects: Online competition and pricing behaviors. *Jackson Hole Economic Symposium Conference Proceedings (Federal Reserve Bank of Kansas City)*.

Chen, Boxiao, Xiuli Chao. 2019. Parametric demand learning with limited price explorations in a backlog stochastic inventory system. *IISE Transactions* **51**(6) 605–613.

Chen, Xi, Yining Wang, Yuan Zhou. 2020. Dynamic Assortment Optimization with Changing Contextual Information. *Journal of Machine Learning Research* **21**.

Cheung, Wang Chi, David Simchi-Levi, He Wang. 2017. Dynamic pricing and demand learning with limited price experimentation. *Operations Research* **65**(6) 1722–1731.

den Boer, Arnoud V. 2015. Dynamic pricing and learning: Historical origins, current research, and new directions. *Surveys in Operations Research and Management Science* **20**(1) 1–18.

Dube, Arindrajit, Ben Zipperer. 2015. Pooling multiple case studies using synthetic controls: An application to minimum wage case studies. IZA Discussion Paper No. 8944.

Elshiewy, Ossama, Daniel Guhl, Yasemin Boztug. 2017. Multinomial Logit Models in Marketing - From Fundamentals to State-of-the-Art. *Marketing ZFP* **39**(3) 32–49.

Ferreira, Kris Johnson, Bin Hong Alex Lee, David Simchi-Levi. 2016. Analytics for an Online Retailer: Demand Forecasting and Price Optimization. *Manufacturing & Service Operations Management* **18**(1) 69–88.

Ferreira, Kris Johnson, David Simchi-Levi, He Wang. 2018. Online Network Revenue Management Using Thompson Sampling. *Operations Research* **66**(6) 1586–1602.

Fisher, Marshall, Santiago Gallino, Jun Li. 2018. Competition-Based Dynamic Pricing in Online Retailing: A Methodology Validated with Field Experiments. *Management Science* **64**(6) 2496–2514.

Garbarino, Ellen, Olivia F. Lee. 2003. Dynamic pricing in internet retail: effects on consumer trust. *Psychology & Marketing* **20**(6) 495–513.

Haws, Kelly L., William O. Bearden. 2006. Dynamic pricing and consumer fairness perceptions. *Journal of Consumer Research* **33**(3) 304–311.

Ho, Daniel E, Kosuke Imai. 2006. Randomization Inference With Natural Experiments: An Analysis of Ballot Effects in the 2003 California Recall Election. *Journal of the American Statistical Association* **101**(475) 888–900.

Hosanagar, Kartik, John Chuang, Ramayya Krishnan, Michael D. Smith. 2008. Service adoption and pricing of content delivery network (cdn) services. *Management Science* **54**(9) 1579–1593.

Huber, Joel, Klaus Zwerina. 1996. The Importance of Utility Balance in Efficient Choice Designs. *Journal of Marketing Research* **33**(3) 307.

Iyengar, Raghuram, Kamel Jedidi, Skander Essegaier, Peter J. Danaher. 2011. The impact of tariff structure on customer retention, usage, and profitability of access services. *Marketing Science* **30**(5) 820–836.

Javanmard, Adel, Hamid Nazerzadeh, Simeng Shao. 2020. Multi-Product Dynamic Pricing in High-Dimensions with Heterogenous Price Sensitivity. Working paper.

Kannan, P.K., Barbara Kline Pope, Sanjay Jain. 2009. Practice prize winner: Pricing digital content product lines: A model and application for the national academies press. *Marketing Science* **28**(4) 620–636.

Kanninen, Barbara J. 2002. Optimal Design for Multinomial Choice Experiments. *Journal of Marketing Research* **39**(2) 214–227.

Kreif, Noemi, Richard Grieve, Dominik Hangartner, Alex James Turner, Silviya Nikolova, Matt Sutton. 2016. Examination of the synthetic control method for evaluating health policies with multiple treated units. *Health Economics* **25** 1514–1528.

Miao, Sentao, Xiuli Chao. 2020. Dynamic joint assortment and pricing optimization with demand learning. *Manufacturing & Service Operations Management* Forthcoming.

Misra, Kanishka, Eric M. Schwartz, Jacob Abernethy. 2019. Dynamic online pricing with incomplete information using multiarmed bandit experiments. *Marketing Science* **38**(2) 226–252.

Newman, Jeffrey P., Mark E. Ferguson, Laurie A. Garrow, Timothy L. Jacobs. 2014. Estimation of choice-based models using sales data from a single firm. *Manufacturing & Service Operations Management* **16**(2) 184–197.

Oh, Min-hwan, Garud Iyengar. 2019a. Multinomial logit contextual bandits. Working paper.

Oh, Min-hwan, Garud Iyengar. 2019b. Thompson sampling for multinomial logit contextual bandits. *Advances in Neural Information Processing Systems*. 3151–3161.

Perakis, Georgia, Divya Singhvi. 2020. Dynamic pricing with unknown non-parametric demand and limited price changes. Working paper.

Petro, G. 2018. All fashion is fast fashion. *Forbes* .

Raghavarao, Damaraju, James B. Wiley, Pallavi Chitturi. 2010. *Choice-based conjoint analysis: models and designs*. CRC Press.

Robbins, Michael W., Jessica Saunders, Beau Kilmer. 2017. A framework for synthetic control methods with high-dimensional, micro-level data: Evaluating a neighborhood-specific crime intervention. *Journal of the American Statistical Association* **112**(517) 109–126.

Sandor, Zsolt, Michel Wedel. 2001. Designing Conjoint Choice Experiments Using Managers' Prior Beliefs. *Journal of Marketing Research* **38**(4) 430–444.

Sandor, Zsolt, Michel Wedel. 2005. Heterogeneous Conjoint Choice Designs. *Journal of Marketing Research* **42**(2) 210–218.

Sauré, Denis, Juan Pablo Vielma. 2018. Ellipsoidal methods for adaptive choice-based conjoint analysis. *Operations Research* **67**(2) 315–338.

Sumida, Mika, Guillermo Gallego, Paat Rusmevichientong, Huseyin Topaloglu, James David. 2021. Revenue-utility tradeoff in assortment optimization under the multinomial logit model with totally unimodular constraints. *Management Science* **67**(5) 2845–2869.

Talluri, K. T., G. J. van Ryzin. 2005. *Theory and Practice of Revenue Management*. Springer-Verlag.

Appendix A Proof of Proposition 1

For any period t, define the random variable $Z_t \in \{0, 1, ..., N_t\}$ to represent the product that a customer shopping assortment t purchases. Its probability mass function given unknown parameters $(\boldsymbol{\beta^f}, \boldsymbol{\beta^p})$ is

$$f(Z_t = i|\boldsymbol{\beta^f}, \boldsymbol{\beta^p}) = q_{it} = \frac{\exp(\boldsymbol{x_{it}^\intercal}\boldsymbol{\beta^f} - \boldsymbol{x_{it}^\intercal}\boldsymbol{\beta^p} p_{it})}{\sum_{l=0}^{N_t} \exp(\boldsymbol{x_{lt}^\intercal}\boldsymbol{\beta^f} - \boldsymbol{x_{lt}^\intercal}\boldsymbol{\beta^p} p_{lt})} \quad \forall i = 0, 1, ..., N_t. \tag{15}$$

We will first determine the Fisher information $I_{Z_t}(\boldsymbol{\beta^f}, \boldsymbol{\beta^p})$ in Z_t, and then extend this to multiple customer arrivals and assortments. Note that since $(\boldsymbol{\beta^f}, \boldsymbol{\beta^p}) \in \mathbb{R}^{2d}$, Fisher information will be represented as a $2d$-square matrix:

$$I_{Z_t}(\boldsymbol{\beta^f}, \boldsymbol{\beta^p}) = \sum_{i=0}^{N_t} \left[q_{it} \left(\frac{\partial \ln f(Z_t = i|\boldsymbol{\beta^f}, \boldsymbol{\beta^p})}{\partial(\boldsymbol{\beta^f}, \boldsymbol{\beta^p})} \right) \left(\frac{\partial \ln f(Z_t = i|\boldsymbol{\beta^f}, \boldsymbol{\beta^p})}{\partial(\boldsymbol{\beta^f}, \boldsymbol{\beta^p})} \right)^\top \right]. \tag{16}$$

We have

$$\ln f(Z_t = i|\boldsymbol{\beta^f}, \boldsymbol{\beta^p}) = (\boldsymbol{x_{it}}, -\boldsymbol{x_{it}} p_{it})^\top (\boldsymbol{\beta^f}, \boldsymbol{\beta^p}) - \ln \left[\sum_{l=0}^{N_t} \exp\left(\boldsymbol{x_{lt}}, -\boldsymbol{x_{lt}} p_{lt}\right)^\top (\boldsymbol{\beta^f}, \boldsymbol{\beta^p}) \right], \tag{17}$$

and taking the derivative yields

$$\frac{\partial \ln f(Z_t = i|\boldsymbol{\beta^f}, \boldsymbol{\beta^p})}{\partial(\boldsymbol{\beta^f}, \boldsymbol{\beta^p})} = (\boldsymbol{x_{it}}, -\boldsymbol{x_{it}} p_{it}) - \sum_{l=0}^{N_t} q_{lt}(\boldsymbol{x_{lt}}, -\boldsymbol{x_{lt}} p_{lt}). \tag{18}$$

Substituting into (16) gives us

$$I_{Z_t}(\boldsymbol{\beta^f}, \boldsymbol{\beta^p}) = \\ \sum_{i=0}^{N_t} \left[q_{it} \left((\boldsymbol{x_{it}}, -\boldsymbol{x_{it}} p_{it}) - \sum_{l=0}^{N_t} q_{lt}(\boldsymbol{x_{lt}}, -\boldsymbol{x_{lt}} p_{lt}) \right) \left((\boldsymbol{x_{it}}, -\boldsymbol{x_{it}} p_{it}) - \sum_{l=0}^{N_t} q_{lt}(\boldsymbol{x_{lt}}, -\boldsymbol{x_{lt}} p_{lt}) \right)^\top \right]. \tag{19}$$

For period t, we observe m_t random samples from Z_t. Since the Fisher information in a random sample of size n is simply n times the Fisher information of a single observation, the Fisher information matrix for period t is $I_t(\boldsymbol{\beta^f}, \boldsymbol{\beta^p}) = m_t I_{Z_t}(\boldsymbol{\beta^f}, \boldsymbol{\beta^p})$. Similarly, the Fisher information matrix over τ periods is

$$I(\boldsymbol{\beta^f}, \boldsymbol{\beta^p}) = \sum_{t=1}^{\tau} m_t I_{Z_t}(\boldsymbol{\beta^f}, \boldsymbol{\beta^p}) = \\ \sum_{t=1}^{\tau} m_t \sum_{i=0}^{N_t} \left[q_{it} \left((\boldsymbol{x_{it}}, -\boldsymbol{x_{it}} p_{it}) - \sum_{l=0}^{N_t} q_{lt}(\boldsymbol{x_{lt}}, -\boldsymbol{x_{lt}} p_{lt}) \right) \left((\boldsymbol{x_{it}}, -\boldsymbol{x_{it}} p_{it}) - \sum_{l=0}^{N_t} q_{lt}(\boldsymbol{x_{lt}}, -\boldsymbol{x_{lt}} p_{lt}) \right)^\top \right]. \tag{20}$$

Appendix B Randomization Inference with Fisher's Exact Test

To measure the significance of our monthly revenue treatment effects, we performed Randomization Inference with Fisher's Exact Test to quantify the probability of observing our monthly treatment effects under the null hypotheses that our algorithm had no effect on revenue each month. Please refer to Ho and Imai (2006) for details on this test. In what follows, we provide the intuition behind the test and then explain the details for how we performed the test to evaluate our field experiment.

If the null hypotheses that our algorithm had no impact on each month's revenue were true, then the percent increase in average daily revenue that we observed (-6.7% for month 1, 14.1% for month 2, and 18.2% for month 3) would be likely to occur due to chance. In other words, if the null hypotheses were true, it should make no difference which studios were labeled as "treatment studios" and which studios were labeled as "control studios" since the treatment of implementing the algorithm had no impact on revenue. Thus, we will proceed with Fisher's Exact Test by repeatedly randomly labeling studios as treatment or control studios and calculating the percent increase in average daily revenue between the two groups for each month. This will create an empirical distribution for each of the monthly revenue treatment effects under the null hypotheses, R_m for months $m = 1, 2, 3$ in the treatment period, and we can determine where along that distribution lies the value of our true monthly treatment effect R_m^* - using the correctly labeled treatment and control studios - providing us with p-values for the probability that the revenue treatment effect would exceed what was observed due to chance. This test is *exact* in the sense that it does not depend on large sample approximation and is *distribution-free* because it does not depend on any distributional assumptions.

Let $j = 1, ..., 50,000$ index the random labeling of studios to treatment vs. control groups, and let $\mathcal{T}_j$ and $\mathcal{C}_j$ represent the sets of studios labeled as the treatment and control groups for the random labeling j, respectively. For each random labeling j, we averaged the daily revenue across all studios $s \in \mathcal{T}_j$ for each of the six months in the pre-period, and identified a synthetic control using studios $s \in \mathcal{C}_j$ whose revenue in each of the six pre-period months closely matched that of the average of studios in $\mathcal{T}_j$. Define w_s^j to be the resultant synthetic control weight calculated via (10) assigned to each studio $s \in \mathcal{C}_j$. We can then calculate the revenue treatment effect for each random labeling j, R_m^j, for each month $m = 1, 2, 3$ in the treatment period as

$$R_m^j = \frac{\frac{1}{|\mathcal{T}_j|}\sum_{s\in\mathcal{T}_j} r_{sm} - \sum_{s\in\mathcal{C}_j} w_s^j r_{sm}}{\sum_{s\in\mathcal{C}_j} w_s^j r_{sm}} \ . \tag{21}$$

Note that this is the same procedure we used to estimate the monthly treatment effects for the true experiment. In order to be consistent with the true randomization process, we discarded all random labelings j for which the synthetic control over the pre-period was poor, resulting in a loss of less than 5% of all random labelings.

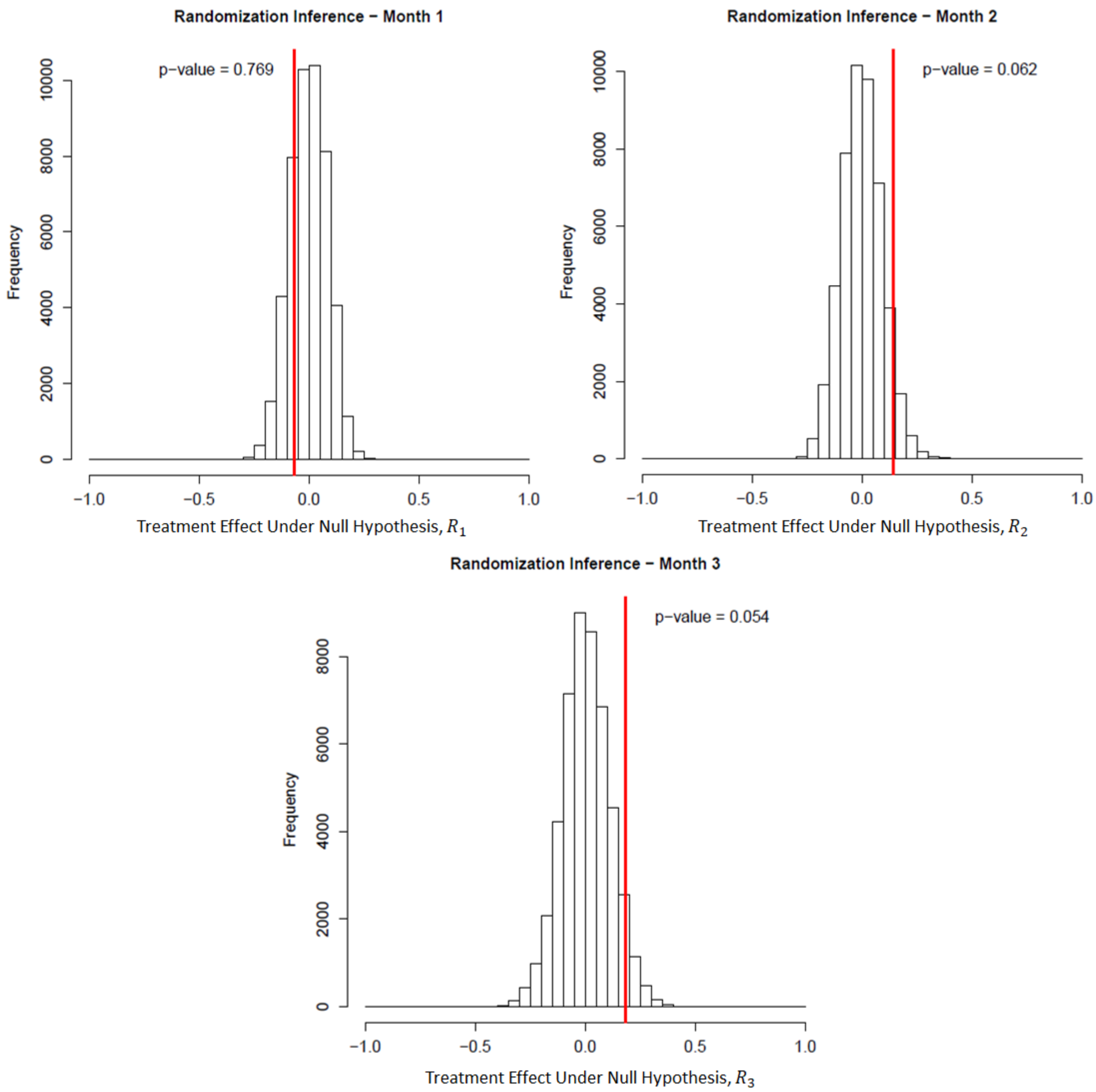

Figure 7 **For each month $m = 1, 2, 3$ in the treatment period, Randomization Inference results using Fisher's Exact Test show the empirical distributions of revenue treatment effects under the null hypotheses, R_m, with the actual treatment effects, R^*_m, represented by red vertical lines. The one-sided p-value shown for each month is empirically calculated as $Pr(R_m \geq R^*_m)$.**

Figure 7 plots a histogram of the values of R^j_m for months $m = 1, 2, 3$ in the treatment period, which can be interpreted as the empirical distribution of R_m for each month. As we would expect under the null hypotheses, $\mathbb{E}[R_m] = 0$ for $m = 1, 2, 3$, since the random labeling of studios to treatment and control groups should have no impact on average daily revenue. We can use this empirical distribution of R_m to calculate a one-sided p-value for each month to measure the significance of our monthly revenue treatment effects. Specifically, for month $m = 1, 2, 3$, we can empirically calculate the one-sided p-value as $Pr(R_m \geq R^*_m)$; this is the probability that we would observe more than an R^*_m increase in revenue in month m due to chance. From the p-values presented in Figure

7, we conclude that the initial dip in revenue in month 1 is unsurprising under the null hypothesis, whereas we have sufficient evidence to reject the null hypotheses in months 2 and 3 at the 10% significance level and conclude that our algorithm had a strong positive effect on revenue.

Appendix C Studio-Specific Synthetic Controls

Figure 8 shows the synthetic control weights, w^*_{st}, for each treatment studio; treatment studios are ordered by the quality of their synthetic control which goes hand-in-hand with the order in which they were selected in the iterative approach described in Section 4.2. As the iterative approach selects studios for the treatment group, the quality of synthetic control degrades. To illustrate this, Figure 9 shows the percent difference in average daily revenue between the treatment studio and its synthetic control for each of the six pre-period months for treatment studios 1-3 (the three highest-quality matches). Similarly, Figure 10 illustrates the percent difference in average daily revenue between the treatment studio and its synthetic control for each of the six pre-period months for treatment studios 21-23 (the three lowest-quality matches); the quality of matches for these studios is much worse. Specifically, for each treatment studio $t = 1, 2, 3, 21, 22, 23$ and pre-period month m, Figures 9 and 10 show

$$\%Diff_{tm} = \frac{r_{tm} - \sum_{s\in\mathcal{C}} w^*_{st} r_{sm}}{\sum_{s\in\mathcal{C}} w^*_{st} r_{sm}} \ . \tag{22}$$

Note that the quality of the synthetic control is much higher when using the average daily revenue over all treatment studios as our treatment unit, as we did in Sections 4.2 and 4.3; the last column of Figure 8 shows the synthetic control weights w^*_s $\forall s \in \mathcal{C}$ for this average treatment unit.

	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_10	T_11	T_12	T_13	T_14	T_15	T_16	T_17	T_18	T_19	T_20	T_21	T_22	T_23	Average Treatment Unit
C_1	0	0	0.05	0	0	0.4	0	0	0	0.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0.02
C_2	0	0	0.01	0.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0.13	0	0	0	0	0	0.01
C_3	0	0.05	0.01	0	0	0	0	0	0.41	0.02	0	0	0.2	0.16	0	0	0	0	0	0	0	0	0.19	0.02
C_4	0.01	0.5	0.01	0	0	0	0.08	0	0	0.01	0	0	0.53	0.41	0	0.46	0	0	0.72	0	0	0	0	0.13
C_5	0	0	0.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0	0.13	0	0	0	0.01
C_6	0	0	0.02	0	0	0	0	0.19	0	0	0	0	0	0	0	0	0.15	0.09	0	0.13	0.2	0	0.11	0.01
C_7	0	0.18	0.26	0.33	0.2	0.09	0	0	0	0.05	0.13	0	0.01	0	0.06	0.04	0	0	0.07	0	0.27	0	0.41	0.27
C_8	0	0.01	0.02	0	0	0	0	0	0.01	0.04	0	0	0.17	0	0	0	0	0	0	0	0	0	0	0.02
C_9	0	0	0.01	0	0	0	0.49	0.55	0.18	0.02	0	0	0	0.04	0	0	0.3	0	0	0	0	0	0.27	0.02
C_10	0	0	0	0	0	0	0	0	0	0.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0.02
C_11	0.03	0	0	0	0	0	0	0.11	0	0	0.12	0	0	0	0.24	0.26	0.45	0	0	0.22	0	0.68	0	0.02
C_12	0	0	0	0	0	0	0	0	0	0.01	0	0.33	0	0	0	0	0	0	0	0	0	0	0	0.01
C_13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.01
C_14	0	0	0	0	0	0	0	0	0	0.01	0.12	0	0	0	0.11	0	0	0	0	0	0	0	0	0.03
C_15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.42	0	0	0	0.06	0	0.01
C_16	0	0	0.01	0	0	0	0	0	0	0.01	0	0	0	0	0	0	0	0	0	0	0	0.12	0	0.02
C_17	0.09	0.13	0.01	0	0.53	0	0	0	0	0.26	0.39	0.11	0	0	0.43	0.24	0	0	0.03	0	0	0	0	0.02
C_18	0.23	0	0.01	0.09	0	0	0	0	0	0.01	0.24	0	0	0	0.13	0	0	0.36	0	0	0	0	0	0.02
C_19	0	0	0	0	0	0	0	0	0	0.01	0	0	0	0	0	0	0	0	0	0.27	0	0	0	0
C_20	0	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0.01
C_21	0	0	0.01	0	0	0	0	0	0	0.01	0	0	0	0	0.03	0	0	0	0	0	0	0	0	0.03
C_22	0.33	0	0	0	0	0	0	0	0	0.01	0	0.06	0	0	0	0	0	0	0	0	0	0	0	0.02
C_23	0	0	0.01	0	0	0	0	0	0	0.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0.02
C_24	0	0	0.01	0	0	0	0	0	0	0.46	0	0	0.1	0.28	0	0	0	0	0	0	0	0	0.02	0.01
C_25	0	0	0.02	0.12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.09	0.14	0	0.01
C_26	0	0.11	0.01	0	0	0.14	0.17	0	0.12	0.01	0	0	0	0.1	0	0	0.08	0	0.17	0.24	0	0	0	0.12
C_27	0	0	0.01	0	0	0.12	0.16	0.15	0	0	0	0	0	0	0	0	0	0	0	0	0.43	0	0	0.04
C_28	0.31	0	0	0	0	0	0	0	0	0.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0.01
C_29	0	0	0.49	0.45	0.27	0.24	0.1	0	0.27	0.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0.05

Figure 8 For each control studio $s \in \mathcal{C}$, synthetic control weights w^*_{st} for each treatment studio t; for example, the synthetic control weight on control studio 7 for treatment studio 2 is 0.18. The last column presents synthetic control weights w^*_s $\forall s \in \mathcal{C}$ for the average treatment unit.

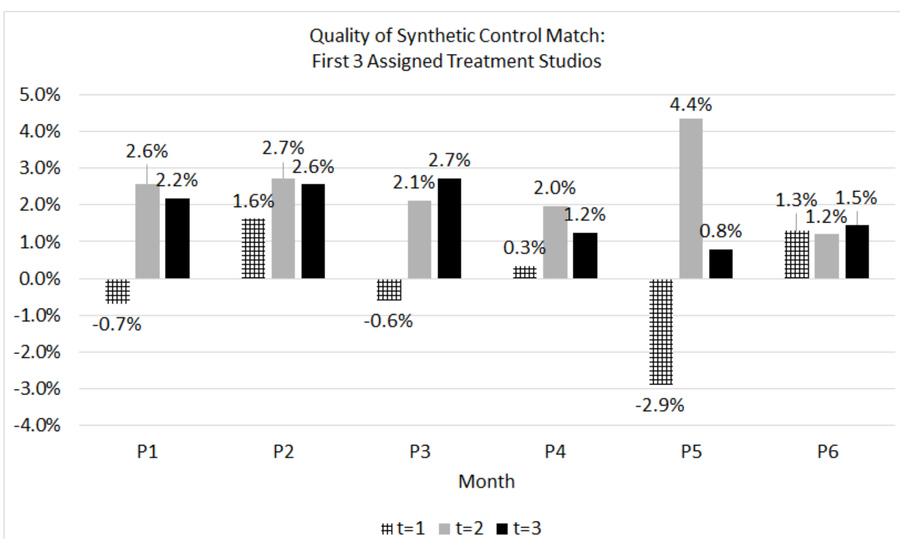

Figure 9 Percent difference in average daily revenue, $\%Diff_{tm}$, over the 6-month pre-period for the first three assigned treatment studios, $t = 1, 2, 3$.

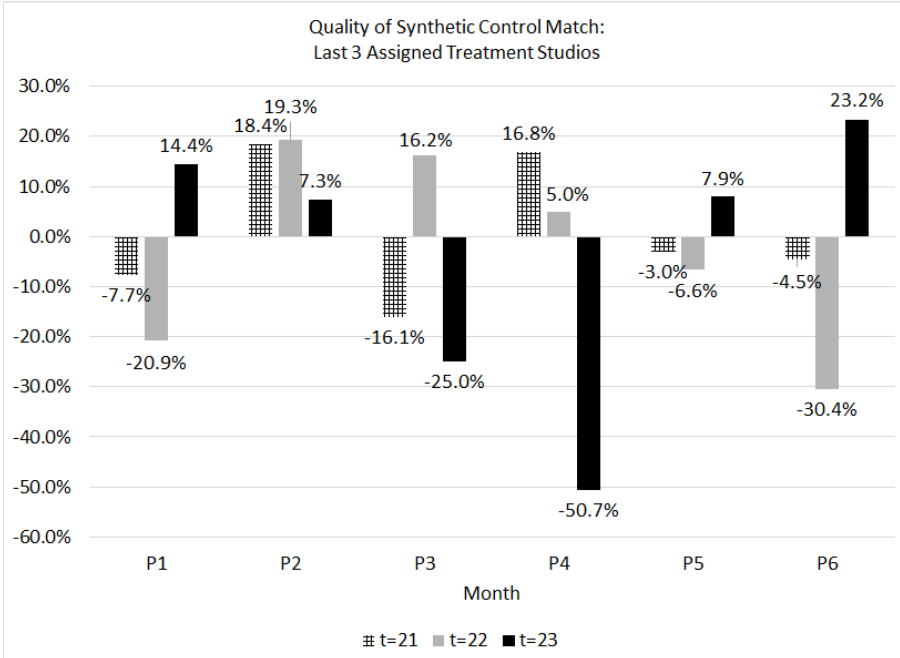

Figure 10 **Percent difference in average daily revenue, $\%Diff_{tm}$, over the 6-month pre-period for the last three assigned treatment studios, $t = 21, 22, 23$.**