Learning Buyer Behavior Under Realistic Pricing Restrictions

Debjyoti Saharoy and **Theja Tulabandhula**
University of Illinois at Chicago

Abstract

We propose a new sample efficient algorithm to learn the parameters governing the purchasing behavior of a utility maximizing buyer, who responds to prices, in a repeated interaction setting. The key feature of our algorithm is that it can work with arbitrary price constraints that the seller may impose. This overcomes a major shortcoming of previous approaches, which use unrealistic prices to learn these parameters making them unsuitable in practice. Once the relevant parameters are learned by our algorithm, in addition to being useful for decisions such as pricing goods to maximize the seller's profit, the models can also be used for other decision problems such as inventory control and promotion planning.

1 Introduction

Modeling the arrival and response behavior of a buyer to a collection of items sold by a seller has a rich history in operations management(Cohen, Lobel, and Leme 2016) and machine learning (Kleinberg and Leighton 2003; Amin, Rostamizadeh, and Syed 2014), and helps answer questions such as: what assortment of items should a seller show a prospective buyer? How should she price them? Much work in this area can be divided into two categories: in the first category, learning the purchase models explicitly is the main goal, and in the second category, maximizing the revenue or some other function given a behavior model is the main objective.

In particular, in the latter category, instead of learning the buyer behavior, one optimizes what is known as *regret*, which is the difference between what the seller could have done in hindsight compared to what they did in a sequence of interactions with the buyer. Although the regret setting is appealing, the techniques and the corresponding algorithms tend to be very specialized (except for perhaps the simplest cases). In particular, many of the general purpose algorithms (such as Thompson Sampling and UCB) depend linearly on the number of actions, which is not-ideal when the action space is large or infinite (as is the case for us). Specialized analysis or algorithms address this dependence issue but depend heavily on the structure of the objective. Further, if the objective changes, either because of business considerations or as new business logic is introduced, one has to design new algorithms and start from scratch. Thus, it is still economical and convenient to decouple the estimation problem from the decision making problem and explicitly estimate the parameters of buyer behavior first. This problem is also called the problem of *pure exploration*.

There is a recent line of work on *learning the behavior of buyers* online (Balcan et al. 2014; Beigman and Vohra 2006; Bei et al. 2016). Compare to these works, our algorithm does not share a key shortcoming, which is the necessity of posting unrealistic prices in the process of learning. Note that, learning buyer behavior in the offline (batch data) setting has also been addressed in recent works. For instance, in (Letham, Letham, and Rudin 2016), the authors learn the parameters of a particular buyer behavior model that considers preference lists. We believe the online setting is relatively more interesting because there is scope for real-time personalization tailored to each individual buyer compared to the offline setting.

In this paper we consider a buyer behavior where only the buyer's objective is sensitive to prices. This type of sensitivity to prices to prices has been considered in the online setting (Roth, Ullman, and Wu 2016) in the context of regret minimizing profit maximization, as well as in the offline setting (Letham, Letham, and Rudin 2016). In each interaction, the seller prices a collection of items and the buyer responds by purchasing various quantities of each item that maximizes her objective/utility. These purchase decisions may be subject to arbitrary (known) feasibility constraints on the bundle of items. We call this model the *unconstrained utility maximization model* (the word unconstrained is used to denote that there is no price based budget constraint, see Section 4). Previously proposed algorithms resort to posting unrealistic prices to induce the buyer to buy/not-buy certain items. Our algorithm eliminates this shortcoming by learning the buyer behavior while being constrained to post prices from a predefined set of realistic prices, which is provided as an input. This makes it more practical and readily applicable in real scenarios. We also show that the number of interactions with the buyer needed by our algorithm is competitive to the previous state of the art methods. Internally, it exploits the concavity property of the buyer's objective and uses projected gradient descent to shrink an uncertainty ellipsoid around the true buyer model parameters.

We summarize our technical contributions below:

1. The key feature of our algorithm is that it searches for prices that induce purchase of specific target bundles, and

creates hyperplanes corresponding to these price vectors to sequentially split the uncertainty set over the buyer's parameters.

2. We also derive a bound on the number of interactions our algorithm needs for learning the buyer parameters. Reducing the number of interactions is important because while learning, the algorithm is agnostic to the revenue generated.

2 Related Work

In the online setting, there is a rich body of work on pricing for profit maximization (Besbes and Zeevi 2015; Chakraborty, Huang, and Khanna 2009; Alaei 2011; Cai and Daskalakis 2011; Blum et al. 2011), where the estimation problem is typically considered secondary and the estimators of the buyer model are heavily tailored to the subsequent objective being optimized for, limiting their universality and reuse. Although regret minimization is attractive because of the possibility of achieving dual objectives (say profit maximization as well as estimation) with the same number of interactions needed for direct estimation, the corresponding algorithms are extremely fragile and sensitive to the structure of the decision problem. The reader is directed to the following references, viz., (Besbes and Zeevi 2015; Chakraborty, Huang, and Khanna 2009; Alaei 2011; Cai and Daskalakis 2011; Blum et al. 2011) for recent results in pricing related regret minimization problems.

One recent work that addresses learning of the unconstrained utility maximization model that we study, is that of (Roth, Ullman, and Wu 2016). There the authors study a profit maximization problem of a leader, who is in a Stackelberg game, when the follower utility function is unknown (but has a parametric form similar to our unconstrained utility maximizing model). They make use of bandit concave optimization and a gradient based method to directly optimize for the prices that control profit. We build on some of their intermediate results (namely that of learning a price that induces a specific bundle of items) for our objective of directly learning the buyer behavior model. Note that their results are not enough to estimate the model paramters completely.

To address this, we also buld on the ellipsoid algorithm (Khachiyan 1980) to get an accurate estimate of the model parameters. Use of ellipsoid uncertainty set has also been previous considered for other online settings. In (Cohen, Lobel, and Leme 2016), the authors study the problem of feature based dynamic pricing using ellipsoidal theory, where the seller receives a single differentiated item in each round that is described by a vector of features. The seller's goal is to design an algorithm that maximizes profit (minimizes regret) when the buyer's valuations for the different item features are not known a priori. Their proposed algorithm essentially does a multidimensional search for the best price to post in each round. Our algorithm can also be seen as performing a multidimensional search, but for parameters related to the entire universe of items. Thus, we work with multiple items in each round in contrast to their single item setting. While using an ellipsoid to represent uncertainty in parameter estimation, the cut direction and the hyperplane placement is straightforward in (Cohen, Lobel, and Leme 2016). On the other hand, in our algorithm, we carefully choose the cut direction as well as position the hyperplane in order to reduce the uncertainty in the model parameters. We do this with realistic constraints on the prices, by solving the dual of a specific optimization problem using projected gradient descent. Finally, note that while their problem is a version of the contextual bandit problem for which general purpose algorithms are already available, our problem is not a contextual bandit instance.

Another way to approach our problem is via regression based methods. As mentioned in (Balcan et al. 2014), such methods tend to need a higher number of interactions with the buyer compared to specialized methods like ours. For instance, in our algorithm we explicitly choose the sequence of query points (adaptive) as well as design specific hyperplanes in such a way so that the uncertainty set around the utility function parameters shrinks the most. These characteristics make our algorithm enjoy stronger query complexity than regression based methods. Note that even regression methods need a subroutine algorithm that we propose in Section 4 (which estimates values of arbitrary bundles via multiple interactions with the buyer).

3 Realistic Prices

Here we define what we mean by realistic prices and motivate it from a practical point of view. These prices will constraint the prices that the seller can set while learning the buyer model (Section 4).

The price p_i of an item i is realistic if it lies within the interval $[p_i^0 - \delta_i, p_i^0 + \delta_i]$, where p_i^0 is the median price point and $2\delta_i$ is the length of the interval (without loss of generality, we can assume symmetry here). This leads to an n-orthotope, which is defined as follows:

Definition 1. *A set of prices is said to be realistic if it is of the following form:*

$$\mathcal{P} = \left\{ p \in \mathbb{R}_+^n \mid \|S^{-1}(p - p^0)\|_\infty \leq 1 \right\}, \qquad (1)$$

where $p^0 \in \mathbb{R}_+^n$ *is the median price point,* $\Delta = [\delta_1 \cdots \delta_n]^T \geq \mathbf{0}$*, and*

$$S = \begin{bmatrix} \delta_1 & & \\ & \ddots & \\ & & \delta_n \end{bmatrix}$$

is its corresponding diagonal matrix. The length of the realistic price interval for each item $i \in [n]$ *is thus* $2\delta_i$*.*

Thus, we consider prices that belong to a transformed $\|\cdot\|_\infty$ norm-ball (scaled by matrix S and translated by p^0). While price covariation constraints are not captured in this definition, suitable affine transformations and other convex bodies such as ellipsoids can also be be used to restrict prices. To obtain closed formulation for algorithmic convergence we assume the set $\mathcal{P}$ is enclosed in a euclidean ball of radius R.

The practical motivation for such a constraint to be imposed while learning the buyer behavior model is straightforward: prices of items in many commercial settings are

only allowed to vary between realistic lower and upper bounds (for instance, setting price as 0 or ∞ is impractical). This is because of business constraints and because of prior knowledge on the market value of goods and items. More involved constraints include bundle prices (where prices are tied to each other) and promotion/discount prices that are also specified by business rules. Further, all such constraints can vary arbitrarily over time. We do not make any strong assumptions on their description and only consider them as exogenous inputs for our algorithm.

Restricting prices has a profound impact on algorithms that have been previously proposed in the literature. For instance, Algorithm 2 in (Balcan et al. 2014) (that addresses learning a slightly different variant of the buyer model considered here) breaks down when one restricts the prices to $\mathcal{P}$. As we show in the subsequent section, our algorithm is able to make progress and outputs an uncertainty set that contains the true parameter of the buyer model even when prices are restricted.

4 Buyer Model

We represent a bundle of goods $x \in C \subseteq [0,1]^n$ (where C is the feasible set) by a vector representing what fraction of each of the n goods is purchased. The prices are represented by a vector $p = (p_1, \cdots, p_n) \in \mathbb{R}^n$. The price of a bundle x is simply $p^T x = \sum_{i=1}^n p_i \cdot x_i$. When the buyer is provided with a price vector p, she buys the *tie-breaking* utility maximizing bundle, $x^*(p)$, which is the optimal solution of the following optimization program:

$$x^*(p) \;=\; \arg\max_{x \in C} \quad U(x) + \tfrac{4}{\mu}\left(\sum_{i=1}^n \sqrt{x_i}\right) - p^T x. \tag{2}$$

Ideally, a utility maximizing buyer would maximize $U(x) - p^T x$, where $U : [0,1]^n \to \mathbb{R}$ specifies their utility for each possible bundle. Since this could potentially lead to multiple optimal bundles (e.g., when U is not strictly concave), we add a tie-breaking perturbation to the original utility function to introduce consistency in the buyer's decision making process. That is, we model the buyer's effective utility function as $U'(x) = U(x) + \frac{4}{\mu}\left(\sum_{i=1}^n \sqrt{x_i}\right)$, where μ is a positive constant. There is nothing special about the choice of the tie breaking function, and many other choices can also be used to make the solution unique (for instance, we can use the Cobb-Douglas function as well). The solution $x^*(p)$ is called the induced bundle at prices p. The seller's goal is to learn the parameters of the function $U(.)$ by observing the bundles bought in a sequence of interactions, where the seller chooses *realistic* prices of items in each interaction.

Assumptions:

We assume that the seller knows the set C of feasible bundles. This is a mild condition, and can be mined from historical purchase data. We also assume that the seller does not know the exact tie breaking parameter μ that the buyer uses, but knows an upper and lower bound on it i.e., $\mu \in [\mu_1, \mu_2]$. We assume tie-breaking to be the only effect of such a function and that its functional form is known beforehand.

To ensure computational tractability of the buyer's problem in Equation (2), we make some generic assumptions about the buyer's utility function. Namely, we assume $U(.)$ is concave on the feasible set C. Also, let $U(x)$ for each $x \in C$ be non-negative and non-decreasing. Since the tie breaking perturbation $\frac{4}{\mu}\left(\sum_{i=1}^n \sqrt{x_i}\right)$ is also non-negative and non-decreasing, so $U'(x)$ non-negative and non-decreasing. The set $C \subseteq \text{dom } U'$ is assumed to be non-empty, compact and convex and $\forall x \in C$, $\|x\|_1 \leq \gamma_1$ and $\|x\|_2 \leq \gamma_2$ (here $\|a\|_q$ refer to the ℓ_q-norm of vector a).

Further, since $U(x)$ is concave on C, $U'(x) = U(x) + \frac{4}{\mu}\left(\sum_{i=1}^n \sqrt{x_i}\right)$, is $\frac{1}{\mu}$-strongly concave (see Proposition 14 in Appendix B) on the set C with respect to $\|\cdot\|_2$ norm. In other words, the buyer's problem defined in Equation (2) is a maximization of a strongly concave function over a convex set C. Hence $x^*(p)$ exists for every $p \in \mathbb{R}^n$ and is unique (follows from strong concavity) as discussed earlier. We also assume that the utility function of the buyer, $U(x)$, is (λ_{val}, β)-Hölder continuous (defined in Appendix B) with respect to the $\|\cdot\|_2$ norm – for all $x, x' \in C$. Thus we have, $|U(x) - U(x')| \leq \lambda_{val} \cdot \|x - x'\|_2^\beta$. Note that this assumption of Hölder-continuity on the utilities is a mild one and is satisfied by a wide range of economically meaningful utilities like Constant Elasticity of Substitution (CES) and Cobb-Douglas utilities.

We restrict the scope of our model to utility functions with linear coefficients and known nonlinearities (these are with respect to x). This includes many concave utility functions (concave in the bundle) including the CES utility function (this is a function of the form $U(x) = (\sum_{i=1}^n \alpha_i x_i^\rho)^\beta$ that has linear coefficients when parameter $\beta = 1$ and $\rho < 1$), the logarithm of the Cobb-Douglas function ($\log U(x) = \sum_{i=1}^n \alpha_i \log x_i$), and any other function that is approximable by a positive polynomial of bundle x. Thus, utilities such as Separable Piecewise-Linear Concave (SPLC), CES, Cobb-Douglas or Leontief functions can also be learned in our setting (although their representation has to be transformed so that the function is linear in the parameters). Later on, without loss of generality, we will assume $U(x) = a^T x = \sum_{i=1}^n a_i \cdot x_i$, with $a \in \mathbb{R}_+^n$.

Note that without an interesting feasible set C of bundles, the learning problem in our setting decomposes into n scalar learning problems that can be solved using binary search. On the other hand, when we have a non-trivial C or a coupling across items through the function U, then binary search is no longer applicable.

The Learning Algorithm

Overview: The algorithm that we propose for learning the unknown parameter vector a^* is based on maintaining uncertainty ellipsoids around a^* and successively shrinking their volume by constructing specific separating hyperplanes (based on observed purchases). At each round t, we start

with an uncertainty ellipsoid E_t and shrink it to get E_{t+1}. In particular, based on the interaction between the buyer and the seller in the current round, we cut E_t with a hyperplane into two regions. And then we update E_{t+1} as the Löwner-John ellipsoid of one of these regions. Appendix A describes the details of some calculus involving ellipsoids, which we use below.

The main technical part of our algorithm is that it works by seeking a desired purchase vector in each round. The purchase vector is then used to deduce a hyperplane that cuts the uncertainty set. Now, this purchase vector cannot be directly accessed as we can only control prices to induce purchase. Below, we show how to use gradient descent and duality to find prices that induce desired bundles.

Along with the price that induces desired bundles, we are able to get the value of these bundles. We compare these values with the minimum and maximum values that are possible given our current uncertainty set over parameter vector a^* and define appropriate hyperplanes to split the uncertainty sets, thus shrinking them.

Finding a price that induces a specific bundle: Consider the following convex program:

$$\begin{array}{ll} \max_{\mathbf{x}\in C} & U'(x) \\ \text{s.t} & x_j \leq \widehat{x}_j \quad \text{for every item } j \in [n], \end{array} \tag{3}$$

where $\hat{x} \in C$ is a specific bundle. Since the utility function $U'(x)$ is non-decreasing and $1/\mu$-strongly concave, we can see that $\widehat{x}$ is the unique optimal solution of the problem in Equation (3). The partial Lagrangian of this formulation (3) is defined as: $\mathcal{L}(x,p) = U'(x) - p^T x + p^T \widehat{x}$, where $p \in \mathbb{R}^n_+$ is the dual variable. We define the Lagrange dual function $g : \mathbb{R}^n \to \mathbb{R}$ to be $g(p) = \max_{x\in C} \mathcal{L}(x,p) = \max_{x\in C} U'(x) - p^T x + p^T \widehat{x}$. Now the dual of the convex program in Equation (3) can be defined as:

$$\begin{array}{ll} \min & g(p) \\ \text{s.t} & p \in \mathbb{R}^n_+. \end{array} \tag{4}$$

Our algorithm (described later) needs to choose a specific bundle $\widehat{x}$ and learn its value $U'(\widehat{x})$. Since we can only control prices, we show how to learn the value $U'(\widehat{x})$ by working with the dual problem. In other words, to compute $U'(\widehat{x})$, which we otherwise could not have since $U'(.)$ is unknown, we define the problem in (3) such that its optimal solution is $\widehat{x}$ itself. We can compute the minimizer $\widehat{p}$ of its dual in (4) because we can control prices. And, by strong duality, we will get $g(\widehat{p}) = U'(\widehat{x}) = \text{OPT}$.

Now we focus on the problem of minimizing the function g, which is also unknown (since $U'(.)$ is unknown). However, due to the structure of the dual problem, the function $g(p)$ can be approximately optimized using a first order optimization technique such as projected gradient descent (refer to Appendix C). In particular, this is the structure we exploit: we have access to the gradients of g and these turn out to be functions of $\widehat{x}$ and the actual bundles purchased by the buyer. Thus, we can set a price p, interact with the buyer to observe the bundle purchased $x^*(p)$ and get access to the gradient. Formally, the following Lemma 2 shows that the bundle $x^*(p)$ purchased by the buyer gives a gradient of the Lagrange dual function $g(\cdot)$ at p.

Lemma 2. *Since the convex program in Equation (3) has a unique optimal solution, therefore $g(p)$ is differentiable at each $p \in \mathcal{P}$. Moreover, if a price vector p induces bundle $x^*(p)$, then the gradient of $g(p)$ at p is given by $\nabla g(p) = \widehat{x} - x^*(p)$.*

Next, we focus on the restriction to realistic prices. We are constrained to set prices only from the realistic price space $\mathcal{P}$, so we can only solve a restricted version of the dual program in Equation (4), which we denote as $\min_{p\in\mathcal{P}} g(p)$. The following Lemma shows that instead of minimizing $g(p)$ in Equation (4) over $p \in \mathbb{R}^n_+$, if it is minimized over the *realistic* price space $\mathcal{P}$ as defined in Definition 1, then the optimal value remains close to OPT.

Lemma 3. *There exists a value R-OPT such that $\min_{p\in\mathcal{P}} g(p) = $ R-OPT. Moreover, $U'(\widehat{x}) \leq $ R-OPT $\leq U'(\widehat{x}) + \tau$, where*

$$\tau = \max\left\{\lambda_{val}\left(\frac{2\underline{L}\gamma_1}{\overline{L}}\right)^{\beta},\ \lambda_{val}^{\frac{1}{1-\beta}}\left(\frac{2}{\overline{L}}\right)^{\frac{\beta}{1-\beta}}\right\} + \underline{L}\gamma_1, \tag{5}$$

$\overline{L} = \|p^0 + \Delta\|_{-\infty}$ and $\underline{L} = \|p^0 - \Delta\|_{\infty}$ (p^0 and Δ defined in Definition 1).

The dual function $g(p)$ is convex, and the following Lemma 4 further shows that $g(p)$ is also strongly smooth (defined in Appendix B).

Lemma 4. *The function g(p) is μ-strongly smooth with respect to the $\|\cdot\|_2$ norm.*

Convexity and smoothness of $g(p)$ are useful below, where we give a projected gradient descent (refer to Appendix C for a brief description) procedure LEARNVALUE$(\widehat{x}, \tau)$ (Algorithm 1). Given a target bundle $\widehat{x} \in C$ and an error budget τ (this is the same value appearing in Lemma 3), LEARNVALUE$(\widehat{x}, \tau)$ minimizes $g(p)$over the realistic price space $\mathcal{P}$ defined in Definition 1, with an additive error of at most τ.

Theorem 5. *(Main Supporting Result) Assuming $g(p_1)$ is known and that $\tau \geq R^2\gamma_2$,* LEARNVALUE$(\widehat{x}, \tau)$ *(Algorithm 1) can estimate R-OPT to accuracy τ. That is after $T = \frac{50\gamma_2\mu_2}{\tau - R^2\gamma_2}$ interactions with the buyer,*

$$g(p_{_T})' - \text{R-OPT} \leq \tau,$$

where $g(p_{_T})'$ is the estimate of R-OPT returned by LEARNVALUE$(\widehat{x}, \tau)$.

Proof. The value of g at each each of the subsequent iterates of the projected gradient procedure can be approximated using Lagrange first order approximation. Thus,

$$g(p_{t+1}) = g(p_t) + \nabla g(p_t)(p_{t+1} - p_t) + \mathcal{E}_{t+1}, \quad t \in [T-1],$$

Algorithm 1 Solving the Lagrangian Dual

procedure LEARNVALUE($\widehat{x}$,τ)
  Initialize: p_1 and $T = \frac{50\gamma_2\mu_2}{\tau - R^2\gamma_2}$.
  for $t = 1, \cdots, T$ **do**
    Observe the purchased bundle, $x^*(p_t)$,by the buyer.
    Update the price vector with projected gradient descent:

$$p_{t+1} = \prod_{\mathcal{P}} \Big[p_t - \eta_t(\widehat{x} - x^*(p_t)) \Big],$$

    where $\eta_t = \gamma / \|\nabla g(p_t)\|$, and $\gamma = 1/T$
  end for
  return $g(p_T)' = g(p_1) + \sum_{t=1}^{T-1} \nabla g(p_t)(p_{t+1} - p_t)$.
end procedure

where $\mathcal{E}$ is the Lagrangian error. Therefore, adding the values of g at each iteration, the sum telescopes and we get

$$\begin{aligned} g(p_T) &= g(p_1) + \sum_{i=1}^{T-1} \nabla g(p_t)(p_{t+1} - p_t) + \sum_{t=1}^{T-1} \mathcal{E}_{t+1} \\ &= g(p_T)' + \sum_{t=1}^{T-1} \mathcal{E}_{t+1}, \end{aligned}$$

where $g(p_T)'$ is the estimate of $g(p_T)$ returned by LEARNVALUE($\widehat{x}, \epsilon$) (Algorithm 1). Thus,

$$|g(p_T) - g(p_T)'| = |\sum_{t=1}^{T-1} \mathcal{E}_{t+1}| \leq \sum_{t=1}^{T-1} |\mathcal{E}_{t+1}|. \quad (6)$$

Now using Taylor's remainder theorem (Theorem 16 in Appendix D) and the fact that g is μ_2-strongly smooth, $\|\nabla^2 g(p)\|_{\max} \leq \|\nabla^2 g(p)\|_2 \leq \mu_2$ for all $p \in \mathcal{P}$, we have

$$|\mathcal{E}_{t+1}| \leq \frac{\mu_2}{2} \|p_{t+1} - p_t\|_2^2, \quad \forall i = 1, ..., T-1. \quad (7)$$

Now using Equation (7) in Equation (6), we get

$$g(p_T)' - g(p_T) \leq \frac{\mu_2}{2} \sum_{t=1}^{T-1} \|p_{t+1} - p_t\|_2^2. \quad (8)$$

Plugging Equation (8) in the following

$$g(p_T) - \text{R-OPT} \leq \frac{\|p^* - p_1\|_2^2 + T\gamma^2}{2\gamma \sum_{t=1}^{T} \frac{1}{\|\nabla g(p_t)\|_2}}, \quad (9)$$

which is the guarantee for projected gradient descent (Theorem 15 in Appendix C) we get:

$$g(p_T)' - \text{R-OPT} \leq \frac{\|p^* - p_1\|_2^2 + T\gamma^2}{2\gamma \sum_{t=1}^{T} \frac{1}{\|\nabla g(p_t)\|_2}} + \frac{\mu_2}{2} \sum_{t=1}^{T-1} \|p_{t+1} - p_t\|_2^2. \quad (10)$$

Therefore, in the Algorithm 1, by choosing constant step lengths in the projected gradient descent procedure i.e., $\eta_t = \gamma / \|\nabla g(p_t)\|$, where $\gamma = 1/T$ we get $\|p_{t+1} - p_t\| = \gamma$ for each $t = [T-1]$. Now, by assuming $\|\nabla g(p_t)\|_2 = \|\widehat{x} - x^*(p)\|_2 \leq \gamma_2$ and $\|p^* - p_1\| \leq R$, Equation (10) becomes:

$$g(p_T)' - \text{R-OPT} \leq \frac{R^2\gamma_2}{2} + \frac{\gamma_2}{2T} + \frac{\mu_2}{2T}.$$

Note that τ as defined in Equation (5) is greater than $R^2\gamma_2$ by assumption. Thus, after at least $T \geq \frac{\gamma_2 + \mu_2}{2(\tau - R^2\gamma_2)}$ iterations the Algorithm 1 produces a solution $g(p_T)'$, which is τ-optimal. □

Therefore, combining Lemma 3 and Theorem 5, we have:

$$U'(\widehat{x}) \leq g(p_T)' \leq U'(\widehat{x}) + 2\tau. \quad (11)$$

Hereafter in this section, for the sake of simplicity of illustration, we focus on learning the buyer's utility function $U(x) = a^{*T}x = \sum_{i=1}^{n} a_i^* \cdot x_i$ assuming it is linear in both the coefficients and the bundle (this is without loss of generality). Hence Equation (11) becomes:

$$a^{*T}\widehat{x} \leq g(p_T)' \leq a^{*T}\widehat{x} + \frac{4}{\mu_1} \left(\sum_{i=1}^{n} \sqrt{\widehat{x}_i} \right) + 2\tau \quad (12)$$

Interval containing the value $U(\widehat{x})$: It turns out that for a target bundle $\widehat{x}$, that the seller has in mind, she can compute an interval $[\underline{b}_t, \overline{b}_t]$ using the uncertainty ellipsoid E_t such that it contains the scalar value $\widehat{x}^T a^*$. Lemma 6 gives the optimum values of the following convex programs:

$$\underline{b}_t = \min_{\tilde{a} \in E(A,c)} \widehat{x}^T \tilde{a}, \quad \text{and } \overline{b}_t = \max_{\tilde{a} \in E(A,c)} \widehat{x}^T \tilde{a}.$$

Lemma 6. *(Grötschel, Lovász, and Schrijver 2012) For any $\widehat{x} \in \mathbb{R}^n \setminus \{0\}$,*

$$arg \max_{\tilde{a} \in E(A,c)} \widehat{x}^T \tilde{a} = c + b, \quad arg \min_{\tilde{a} \in E(A,c)} \widehat{x}^T \tilde{a} = c - b,$$

where $b = A\widehat{x} / \sqrt{\widehat{x}^T A \widehat{x}}$.

So, if $g(p_T)' \leq (\underline{b} + \overline{b})/2 = \widehat{x}^T c$, then the unknown parameter a^* lies in the halfspace

$$H = \{\tilde{a} \in \mathbb{R}^n : \widehat{x}^T \tilde{a} \leq \widehat{x}^T c\}. \quad (13)$$

On the other hand if $g(p_T)' \geq (\underline{b} + \overline{b})/2 = \widehat{x}^T c$, then by Equation (12), the unknown parameter a^* lies in the halfspace

$$H = \left\{ \tilde{a} \in \mathbb{R}^n : \widehat{x}^T \tilde{a} \geq \widehat{x}^T c - \left(\frac{4}{\mu_1} \left(\sum_{i=1}^{n} \sqrt{\widehat{x}_i} \right) + 2\tau \right) \right\}. \quad (14)$$

The main algorithm: Now we present the LEARNUTILITY algorithm (Algorithm 2) for learning the parameter a^* of the buyer's utility function.

Algorithm 2 Learning Utility Maximizing Buyer's Model

procedure LEARN-UTILITY (ϵ)
  $E_0 = E(A_0, c_0) \subseteq \mathbb{R}^n$ is the initial uncertainty ellipsoid with $A_0 = R_a \cdot I$ for $R_a > 0$ as defined in Theorem 9.
  Pick bundle $x_t = \arg\max_{x \in \mathcal{C}} \sqrt{x^T A_0 x}$
  do
    Compute $\underline{b_t}$ and $\overline{b_t}$ using Lemma 6.
    $g(p_T)' \leftarrow$ LEARNVALUE(x_t, τ)
    if $g(p_T)' \leq (\underline{b} + \overline{b})/2 = x_t^T c$ **then** H_t is (13),
    else H_t is (14).
    $E_{t+1} = \text{LJohn}(E_t \cap H_t)$ (here LJohn() finds the Löwner-John ellipsoid of its argument).
    Pick bundle $x_t = \arg\max_{x \in \mathcal{C}} \sqrt{x^T A_{t+1} x}$
  while $(2\sqrt{x_t^T A_{t+1} x_t} > \epsilon)$
end procedure

Analysis

Note that in LEARN-UTILITY , the uncertainty ellipsoid E_{t+1} for the next iteration is updated using the computation $E_{t+1} = \text{LJohn}(E_t \cap H_t)$, where H_t is defined by either Equation (13) or (14). The former induces a central cut in the ellipsoid $E_t(A, c)$, i.e. the hyperplane H_t passes through the center c and eliminates half of the volume of the ellipsoid. On the other hand, the later hyperplane induces a shallow cut and removes less than half of the volume. Without loss of generality (as we only need an upper bound on the number of iterations needed by LEARN-UTILITY to learn a^*), we assume that at each iteration the relevant hyperplane induces a shallow cut. That is, H_t is either of the following:

$$\begin{aligned} H_t &= \{\tilde{a} \in \mathbb{R}^n : x_t^T \tilde{a} \leq x_t^T c + \delta\} \text{ or,} \\ &= \{\tilde{a} \in \mathbb{R}^n : x_t^T \tilde{a} \geq x_t^T c - \delta\}, \end{aligned} \tag{15}$$

where $\delta = \left(\frac{4}{\mu_1}\left(\sum_{i=1}^{n}\sqrt{x_{ti}}\right) + 2\tau\right)$ is the depth of the cut induced. For LEARN-UTILITY to work we need the depth δ to be at most $\frac{\sqrt{x_t^T A_t x_t}}{n}$, i.e. $\delta \leq \frac{\sqrt{x_t^T A_t x_t}}{n}$. As the portion $\frac{4}{\mu_1}\left(\sum_{i=1}^{n}\sqrt{x_{ti}}\right)$ takes effect only in tie-breaking, i.e., we can assume μ_1 to be a large constant. Hence, the constraint on the depth of the shallow cut becomes $\sqrt{x_t^T A_t x_t} \geq 2n\tau$. Also, note that the Algorithm LEARN-UTILITY continues as long as $2\sqrt{x_t^T A_t x_t} > \epsilon$. So the shallow cut condition is met (in other words the algorithm is able to find an x_t in each iteration) as long as $\tau < \epsilon/4n$.

Next, the computation of the Löwner-John ellipsoids of the sets that remain after shallow cuts follows from (Grötschel, Lovász, and Schrijver 2012). The Löwner-John ellipsoid of the set $E_t(A_t, c_t) \cap \{\tilde{a} \in \mathbb{R}^n : x_t^T \tilde{a} \leq x_t^T c + \delta\}$ is $E(A_{t+1}, c_t - \frac{1+n\alpha_t}{n+1} b_t)$, and of the set $E_t(A_t, c_t) \cap \{\tilde{a} \in \mathbb{R}^n : x_t^T \tilde{a} \leq x_t^T c - \delta\}$ is $E(A_{t+1}, c_t + \frac{1+n\alpha_t}{n+1} b_t)$, where $\alpha_t = -\frac{\delta}{\sqrt{x_t^T A_t x_t}}$, $b_t = A_t x_t / \sqrt{x_t^T A_t x_t}$ and A_{t+1} is defined in Appendix A.

In what follows we present the performance guarantee of the Algorithm 2. Firstly, to bound the minimum eigenvalue λ_n at successive iterations of our algorithm, we give the following two lemmas from (Cohen, Lobel, and Leme 2016) also applicable in our setting. In (Cohen, Lobel, and Leme 2016), they are used in the analysis of a different algorithm in a different setting (regret minimization).

Lemma 7. *For any iteration step t, we have $\lambda_n(A_{t+1}) \geq \frac{n^2}{(n+1)^2}\lambda_n(A_t)$.*

Lemma 8. *There exists a sufficiently small $k = k(n)$ such that if $\lambda_n(A_t) \leq k\epsilon^2$ and $x_t^T A_t x_t > \frac{1}{4}\epsilon^2$, then $\lambda_n(A_{t+1}) \geq \lambda_n(A_t)$, i.e., the smallest eigenvalue doesn't decrease after the update. One can assume $k = \frac{1}{400n^2}$.*

Using the above two lemmas, we can show that the number of rounds needed by LEARN-UTILITY is upper bounded.

Theorem 9. *The algorithm* LEARN-UTILITY *terminates after at most $20n^2 \ln\left(\frac{20R_a(n+1)}{\epsilon}\right)$ iterations, where R_a is the radius of the initial uncertainty set E_0.*

Combining Theorem 5 and 9, we get the following bound on the interactions needed to get a tight uncertainty set around the unknown parameter a^* of the buyer's utility function.

Theorem 10. *(Main Result) Assume that the feasible set C, the realistic price set $\mathcal{P}$ and the algorithm parameter ϵ obey the condition: $R^2\gamma_2 \leq \tau \leq \frac{\epsilon}{4n}$. Then, after at most $t \cdot T$ interactions with the buyer, where $t = 20n^2 \ln\left(\frac{20R_a(n+1)}{\epsilon}\right)$ and $T = \frac{50\gamma_2\mu_2}{\tau - R^2\gamma_2}$, algorithm* LEARN-UTILITY *outputs uncertainty set $E(A_t, c_t)$ such that the buyer utility parameter $a^* \in E(A_t, c_t)$ and $\max_{x \in \mathcal{C}} 2\sqrt{x^T A_t x} \leq \epsilon$.*

5 Discussion

We now make a few comparative remarks about our solution, and discuss its limitations and potential for future work. As mentioned before in Section 2, our method builds on two recent previous works, viz., (Roth, Ullman, and Wu 2016) and (Cohen, Lobel, and Leme 2016). In (Roth, Ullman, and Wu 2016), the authors investigate a similar pricing problem with a profit objective where: (1) they are concerned about regret, (2) they have unknown costs for the seller, and (3) they solve a particular Stackelberg game. In (Cohen, Lobel, and Leme 2016): (1) the authors learn a linear buyer valuation function in a regret setting by pricing single items as they arrive online and are described by a finite number of features, and (2) they make use of uncertainty ellipsoids to upper bound the number of suboptimal pricing choices while maximizing profit.

From (Roth, Ullman, and Wu 2016), we re-purpose the use of projected gradient descent based technique (used in solving the convex program in Equation (3)) for interacting with the buyer. While they do not need any specific variant of the gradient descent algorithm, we explicitly choose a certain step rule (constant step length) to bound our learning

errors. Our own contribution here is the use of these gradient descent moderated interactions for splitting the uncertainty ellipsoids, whereas they use such interactions for solving a specific structured Stackelberg game (requiring very different tools and techniques in the process). Similar to (Cohen, Lobel, and Leme 2016), we use two specific eigenvalue lower bounding lemmas (see Lemma 7 and 8) to bound the number of rounds of interaction need by our algorithm, with the key difference being the way the separating hyperplanes are generated in each of our methods. Further, we note that algorithms in both (Roth, Ullman, and Wu 2016) and (Cohen, Lobel, and Leme 2016) cannot be easily extended to the realistic prices setting (defined in Section 3), which is our key emphasis here.

For the buyer models that we consider, the utility $U(x)$ need not be linear in the bundle, so even polynomial utility functions can be learned as long as certain conditions such as concavity and positive coefficients can be met. This makes our algorithm and its analysis in Section 4 more generally applicable. We completely side-step the issue of identifiability of the model in our treatment, by reporting uncertainty sets instead of point estimates of the true parameters. When allowable prices are exogenous, it may happen that the best uncertainty set is still very loose due to stringent pricing restrictions. Another important issue that we did not address here is that of modeling stochasticity in the buyer models. This is a very natural setting to propose algorithms for, and has been addressed in, for instance (Roth, Ullman, and Wu 2016). As our algorithm uses an ellipsoidal search template for which noisy generalizations exist, it can potentially be extended to the noisy case (appropriate noise models have to be specified here). Our algorithm also uses projected gradient descent while interacting with the buyer. Thus, noisy gradient information obtained from the buyer can potentially be dealt with as well. An empirical analysis of the performance of the algorithm with real datasets would be a future direction.

6 Conclusions

In this paper we proposed a sample efficient online method to learn the behavior model of a buyer that maximizes utility without any budget restriction, by controlling prices. One of the key advantages of our algorithm is that it is amenable to exogenous pricing restrictions imposed by business and managerial constraints, making it relatively more practical and user-friendly than previously proposed approaches. Using our algorithm, practitioners can build a model of buyer behavior from purchase and pricing data, which can be subsequently used for inventory, pricing and other business decisions.

A Ellipsoidal Calculus

Here we discuss some technical preliminaries regarding ellipsoids. For more details, we refer the reader to (Kurzhanskiy and Varaiya 2006).

Definition

An $n \times n$ matrix A is symmetric if $A = A^T$, i.e., it is equal to its transposed matrix. A basic linear algebra fact states that every symmetric matrix A admits an eigenvalue decomposition, i.e., we can write $A = QDQ^T$, where Q is a $n \times n$ orthogonal matrix (i.e., $Q^TQ = I$) and D is a diagonal matrix with elements with elements $\lambda_1 \geq \cdots \geq \lambda_n$ in its main diagonal. We refer to $\lambda_i(A)$ as the i^{th} largest eigenvalue of A. A symmetric matrix is said to be positive definite if all of its eigenvalues are strictly positive, i.e., $\lambda_d(A) > 0$.

An ellipsoid E is a subset of $\mathbb{R}^d$ defined by a vector $c \in \mathbb{R}^d$, which we call the center and a positive definite matrix A as follows:

$$E(A, c) := \{x \in \mathbb{R}^d : (x - c)^T A^{-1}(x - c) \leq 1\}.$$

Each of the n radii of $E(A, a)$ corresponds to the square root of an eigenvalue of A and the volume of the ellipsoid is given by:

$$\text{VOL}\left(E(A, c)\right) = V_n \cdot \sqrt{\prod_i \lambda_i(A)},$$

where V_n is a constant that depends only on n and corresponds to the volume of the unit ball in $\mathbb{R}^n$. Since the volume only depends on the matrix A, we will sometimes suppress the center c and write VOL $(E(A))$.

Bounding the Intersection of an Ellipsoid and a Halfspace

The hyperplane perpendicular to a vector $p \in \mathbb{R}^n \setminus \{0\}$ and passing through c is given by $H = \{p'(x - c) = 0\}$. This plane cuts the ellipsoid $E(A, c)$ in two symmetric pieces. The smallest ellipsoid containing either of these pieces (called the Löwner John ellipsoid) can be computed as follows. The smallest ellipsoid containing $E(A, c) \cap \{x \in \mathbb{R}^n : p'(x - c) \leq 0\}$ is $E(\tilde{A}, c - \frac{1}{n+1}b)$ and the smallest ellipsoid containing $E(A, c) \cap \{x \in \mathbb{R}^n : p'(x - c) \geq 0\}$ is $E(\tilde{A}, c + \frac{1}{n+1}b)$, where:

$$\tilde{A} = \frac{n^2}{n^2 - 1}(A - \frac{2}{n + 1}bb'), \text{ and } b = Ap/\sqrt{p'Ap}. \quad (16)$$

A central fact used in the analysis of any ellipsoidal search method is that:

$$\text{VOL}\left(E(\tilde{A})\right) \leq e^{-1/2n} \cdot \text{VOL}\left(E(A)\right). \quad (17)$$

More generally, the hyperplane $H = \{x \in \mathbb{R}^n : p^T x \leq \gamma\}$ intersects the ellipsoid $E(A, c)$ if and only if:

$$-1 \leq \alpha \leq 1, \text{ where } \alpha := \frac{p^T c - \gamma}{\sqrt{p^T Ap}}.$$

The Löwner-John ellipsoid $E(A', c')$ that circumscribes the set $E(A, c) \cap H$ can be determined as follows: if $-1 \leq \alpha \leq -1/n$, then $E(A', c') = E(A, c)$. On the other hand, if $-1/n \leq \alpha \leq 1$, then

$$c' := c - \frac{1 + n\alpha}{n + 1}b$$

$$A' := \frac{n^2}{n^2 - 1}(1 - \alpha^2)(A - \frac{2(1 + n\alpha)}{(n + 1)(1 + \alpha)}bb^T),$$

where b is defined as $b := \frac{1}{\sqrt{p^T Ap}} Ap$. Further generalizations can be found in (Kurzhanskiy and Varaiya 2006).

B Hölder Continuity, Strongly Convex and Smooth Functions

Here we provide definitions for Hölder continuity, strong convexity and smoothness of functions, and state a theorem which shows that strong convexity and smoothness are dual properties.

We consider the function $f : \mathcal{X} \to \mathbb{R}$ has dom $f = \{x | f(x) < \infty\}$.

Definition 11. *A function $f : \mathcal{X} \to \mathbb{R}$ is (λ, β)-Hölder continuous for some constants $\lambda \geq 1$, and $\beta \in (0, 1]$ if for any $x, y \in \mathcal{X}$, $|f(x) - f(y)| \leq \lambda \|x - y\|^\beta$.*

Definition 12. *A function $f : \mathcal{X} \to \mathbb{R}$ is μ-strongly convex with respect to a norm $\|\cdot\|$ if for all x, y in the relative interior of the domain of f and $\alpha \in (0, 1)$ we have*

$$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha) f(y) - \frac{1}{2}\beta\alpha(1-\alpha)\|x - y\|^2.$$

Definition 13. *A function $f : \mathcal{X} \to \mathbb{R}$ is μ-strongly smooth with respect to a norm $\|\cdot\|$ if f is everywhere differentiable and if for all $x, y \in \mathcal{X}$ we have*

$$|f(x) - f(y) - \nabla f(y)^T (x - y)| \leq \frac{\mu}{2}\|x - y\|_2^2.$$

If f is also convex then,

$$0 \leq f(x) - f(y) - \nabla f(y)^T (x - y) \leq \frac{\mu}{2}\|x - y\|_2^2.$$

Proposition 14. *The function f(x) $= \frac{4}{\mu}\left(\sum_{i=1}^n \sqrt{x_i}\right)$ is $\frac{1}{\mu}$strongly-concave, with respect to $\|\cdot\|_2$ norm, over any convex set $C \subseteq (0, 1]^n$.*

C Projected Gradient Descent with Constant Step Length

Here we briefly discuss the projected gradient descent method, with the choice of constant step lengths for the stepping-rule. It is a first order method, i.e., it can be used to find ϵ-optimal solutions of convex optimization problems, given access to the gradients of the objective.

Let $\mathcal{P} \subseteq \mathbb{R}^n$ be a compact and convex set that is contained in an Euclidean ball of radius R, centered at some point $p_1 \in \mathbb{R}^n$. Let $g : \mathbb{R}^n \to \mathbb{R}$ be a convex function. Let $\prod_{\mathcal{P}}$ denote the projection operator onto the space $\mathcal{P}$, such that for any $p \in \mathbb{R}^n$

$$\prod_{\mathcal{P}}(p) = \arg\min_{p' \in \mathcal{P}} \frac{1}{2}\|p - p'\|_2^2.$$

Projected gradient descent is an iterative algorithm that starts at $p_1 \in \mathcal{P}$ and iterates the following equation

$$p_{t+1} = \prod_{\mathcal{P}} (p_t - \eta_t \nabla g(p_t)),$$

where η_t is the step size at iteration t. When $\eta_t = \frac{\gamma}{\|\nabla g(p_t)\|_2}$ (constant step length rule) for some $\gamma > 0$, the algorithm has the following performance guarantee.

Theorem 15. *When g is a convex and $\mathcal{P}$ is a compact and convex set, the projected gradient descent algorithm with $\eta_t = \frac{\gamma}{\|\nabla g(p_t)\|_2}$ satisfies*

$$\|p_{t+1} - p_t\|^2 = \gamma, \text{ and}$$

$$g(p_t) - g^* \leq \frac{R^2 + t\gamma^2}{2\gamma \sum_{u=1}^t \frac{1}{\|\nabla g(p_u)\|_2}}.$$

Let $\|\nabla g(p_t)\|_2 \leq G$ for all t. Then, $g(p_t) - g^* \leq \frac{R^2 + t\gamma^2}{2t\gamma/G}$. Thus, choosing $\gamma < 2\epsilon/G$, after at least $\frac{R^2}{2\gamma\epsilon/G - \gamma^2}$ iterations, the projected gradient descent algorithm produces a solution that is ϵ-close to the optimal.

D Taylor's Theorem for Multivariate Function Approximation

In this section we give the Taylor's theorem for functions of several variables. We start by defining the multi-index notation.

A *multi-index* is an n-tuple of non negative integers. Multi-indices are generally denoted by Greek letters α or β:

$$\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_n), \quad (\alpha_j \in \{0, 1, 2, \cdots\}).$$

If α is a multi-index, then

$$|\alpha| = \alpha_1 + \alpha_2 + \cdots + \alpha_n, \quad \alpha! = \alpha_1! \alpha_2! \cdots \alpha_n!,$$
$$x^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n} \quad (\text{where } x = (x_1, \cdots, x_n) \in \mathbb{R}^n),$$
$$\partial^\alpha f = \partial_1^{\alpha_1} \partial_2^{\alpha_2} \cdots \partial_n^{\alpha_n} f = \frac{\partial^{|\alpha|} f}{\partial x_1^{\alpha_1} \partial x_2^{\alpha_2} \cdots \partial x_n^{\alpha_n}}$$

The number $|\alpha|$ is called the degree of α. Thus, the order of α is the same as the order of x^α as a monomial or the order of ∂^α as a partial derivative. The generic kth-order partial derivative of f can be written simply as $\partial^\alpha f$ with $|\alpha| = k$.

Theorem 16. *Suppose $f : \mathbb{R}^n \to \mathbb{R}$ is of class C^{k+1} on a convex set $\mathcal{S}$. If $a \in \mathcal{S}$ and $x = a + h \in \mathcal{S}$, then the Taylor expansion and remainder for $f(x)$ about the point a, by restricting f to the line joining a and x, is given by*

$$f(a + h) = \sum_{|\alpha| \leq k} \frac{\partial^\alpha f(a)}{\alpha!} h^\alpha + R_{a,k}(h),$$

where the remainder $R_{a,k}(h)$ in Lagrange's form is given by

$$R_{a,k}(h) = \sum_{|\alpha| = k+1} \partial^\alpha f(a + ch) \frac{h^\alpha}{\alpha!} \text{ for some } c \in (0, 1).$$

If $\max_{|\alpha|=k+1} |\partial^\alpha f(x)| \leq M$ for $x \in \mathcal{S}$, then

$$|R_{a,k}(h)| \leq \frac{M}{(k+1)!}\|h\|_2^{k+1}.$$

References

[Akritas 2009] Akritas, A. G. 2009. Linear and quadratic complexity bounds on the values of the positive roots of polynomials. *Journal of Universal Computer Science* 15(3):523–537.

[Alaei 2011] Alaei, S. 2011. Bayesian combinatorial auctions: Expanding single buyer mechanisms to many buyers. In *Proceedings of the Fifty-second Annual IEEE Symposium on Foundations of Computer Science*, 512–521. IEEE Computer Society.

[Amin, Rostamizadeh, and Syed 2014] Amin, K.; Rostamizadeh, A.; and Syed, U. 2014. Repeated contextual auctions with strategic buyers. In *Advances in Neural Information Processing Systems*, 622–630.

[Balcan et al. 2014] Balcan, M.-F.; Daniely, A.; Mehta, R.; Urner, R.; and Vazirani, V. V. 2014. Learning economic parameters from revealed preferences. In *International Conference on Web and Internet Economics*, 338–353. Springer.

[Bei et al. 2016] Bei, X.; Chen, W.; Garg, J.; Hoefer, M.; and Sun, X. 2016. Learning market parameters using aggregate demand queries. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*.

[Beigman and Vohra 2006] Beigman, E., and Vohra, R. 2006. Learning from revealed preference. In *Proceedings of the Seventh ACM Conference on Electronic Commerce*, 36–42. ACM.

[Besbes and Zeevi 2015] Besbes, O., and Zeevi, A. 2015. On the (surprising) sufficiency of linear models for dynamic pricing with demand learning. *Management Science* 61(4):723–739.

[Blum et al. 2011] Blum, A.; Gupta, A.; Mansour, Y.; and Sharma, A. 2011. Welfare and profit maximization with production costs. In *Proceedings of the Fifty-second Annual IEEE Symposium on Foundations of Computer Science*, 77–86. IEEE.

[Boyd and Vandenberghe 2004] Boyd, S., and Vandenberghe, L. 2004. *Convex Optimization*. Cambridge university press.

[Cai and Daskalakis 2011] Cai, Y., and Daskalakis, C. 2011. Extreme-value theorems for optimal multidimensional pricing. In *Proceedings of the Fifty-second Annual IEEE Symposium on Foundations of Computer Science*, 522–531. IEEE.

[Chakraborty, Huang, and Khanna 2009] Chakraborty, T.; Huang, Z.; and Khanna, S. 2009. Dynamic and non-uniform pricing strategies for revenue maximization. In *Proceedings of the Fiftieth Annual IEEE Symposium on Foundations of Computer Science*.

[Cohen, Lobel, and Leme 2016] Cohen, M.; Lobel, I.; and Leme, R. P. 2016. Feature-based dynamic pricing. In *Proceedings of the ACM Conference on Economics and Computation*.

[Grötschel, Lovász, and Schrijver 2012] Grötschel, M.; Lovász, L.; and Schrijver, A. 2012. *Geometric Algorithms and Combinatorial Optimization*, volume 2. Springer Science & Business Media.

[Khachiyan 1980] Khachiyan, L. G. 1980. Polynomial algorithms in linear programming. *USSR Computational Mathematics and Mathematical Physics* 20(1):53–72.

[Kleinberg and Leighton 2003] Kleinberg, R., and Leighton, T. 2003. The value of knowing a demand curve: Bounds on regret for online posted-price auctions. In *Proceedings of the Fourty-fourth Annual IEEE Symposium on Foundations of Computer Science*, 594–605. IEEE.

[Kurzhanskiy and Varaiya 2006] Kurzhanskiy, A. A., and Varaiya, P. 2006. Ellipsoidal toolbox. *EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2006-46*.

[Letham, Letham, and Rudin 2016] Letham, B.; Letham, L. M.; and Rudin, C. 2016. Bayesian inference of arrival rate and substitution behavior from sales transaction data with stockouts. In *Proceedings of the Twenty-second ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, 1695–1704. ACM.

[Obreshkov 1963] Obreshkov, N. 1963. *Verteilung und Berechnung der Nullstellen Reeller Polynome*. Deutscher Verlag der Wissenschaften.

[Roth, Ullman, and Wu 2016] Roth, A.; Ullman, J.; and Wu, Z. S. 2016. Watch and learn: Optimizing from revealed preferences feedback. In *Proceedings of the Forty-eighth ACM Symposium on Theory of Computing*, 949–962. ACM.