Learning the Demand Curve in Posted-Price Digital Goods Auctions

Meenal Chhabra
Rensselaer Polytechnic Inst.
Dept. of Computer Science
Troy, NY, USA
chhabm@cs.rpi.edu

Sanmay Das
Rensselaer Polytechnic Inst.
Dept. of Computer Science
Troy, NY, USA
sanmay@cs.rpi.edu

ABSTRACT

Online digital goods auctions are settings where a seller with an unlimited supply of goods (e.g. music or movie downloads) interacts with a stream of potential buyers. In the posted price setting, the seller makes a take-it-or-leave-it offer to each arriving buyer. We study the seller's revenue maximization problem in posted-price auctions of digital goods. We find that algorithms from the multi-armed bandit literature like UCB, which come with good regret bounds, can be slow to converge. We propose and study two alternatives: (1) a scheme based on using Gittins indices with priors that make appropriate use of domain knowledge; (2) a new learning algorithm, LLVD, that assumes a linear demand curve, and maintains a Beta prior over the free parameter using a moment-matching approximation. LLVD is not only (approximately) optimal for linear demand, but also learns fast and performs well when the linearity assumption is violated, for example in the cases of two natural valuation distributions, exponential and log-normal.

Categories and Subject Descriptors

J.4 [**Social and Behavioral Sciences**]: Economics

General Terms

Algorithms, Economics

Keywords

Electronic markets, Economically-motivated agents, Single agent learning

1. INTRODUCTION

Digital goods auctions are those where a seller with an unlimited supply of identical goods interacts with a population of buyers who desire one unit of that good [12, 11]. These are typically thought of as digital goods which can be produced at negligible cost, for example, rights to watch a movie broadcast, or to download an audio file.

Consider the problem faced by a company that has the rights to a piece of music, and wants to market it to consumers. There is some underlying valuation distribution on the potential population of buyers, reflecting how much each potential buyer values that piece. However, the seller is not aware of this distribution, and can only learn it through interaction with buyers. The seller's goal is to maximize her own revenue. While such problems have typically been dealt with by using a few discrete possible prices and estimating popularity, this has mostly been due to the transaction costs associated with regularly changing prices. Dynamic pricing mechanisms, on the other hand, are increasingly available to sellers, and it is now practical to consider strategies that change prices online [13]. The typical interaction will be that the user searches a music database for the piece, sees a price, and decides whether or not to buy.

In this kind of posted-price mechanism [15, 3], the seller offers a single price, and an arriving buyer has the option to either complete the purchase at that price, or not go through with it. If the seller knew the distribution of valuations, the pricing problem for revenue maximization would be simple to solve, yielding a single fixed price to be offered to all the buyers (under the assumption that the seller has no way of discriminating between buyers, or finding out their individual valuations). This distribution can also be thought of as the demand curve, because an arriving buyer will only buy if her valuation exceeds the posted price being offered.

Posted price mechanisms have also received attention in the context of limited supply auctions [4]. There has been work in economics on learning the demand curve in posted price auctions when the seller has a single unit of the item to sell [5], and also on learning the demand curve using buyers' bidding behavior in non-posted price settings [19].

Posted price auctions in which the seller must learn the demand curve are a natural application for the tools of dynamic programming and reinforcement learning because they exhibit a classic exploration-exploitation dilemma. The quoted price serves as both a profit-seeking mechanism (exploitation) as well as an information-gathering one (exploration). In the context of two-sided posted-price mechanisms in finance where a "market maker" offers to both buy and sell a security at some price, Das and Magdon-Ismail [7] use dynamic programming techniques to show that there are times when it is optimal to make significant losses in order to learn the valuation distribution more quickly. In digital goods auctions the seller does not make a loss, but may lose out on potentially higher revenue instead.

Given the exploration-exploitation dilemma inherent in the problem, it is natural that many of the algorithms analyzed for posted price selling with unknown demand have been based on the multi-armed bandit literature. Several of

Cite as: Learning the Demand Curve in Posted-Price Digital Goods Auctions, Meenal Chhabra and Sanmay Das, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. XXX-XXX.

these schemes have been shown to possess good properties in terms of asymptotic regret for the seller's revenue maximization problem in the unlimited supply setting. Blum *et al* [3] discuss the application of Auer *et al*'s [2] EXP3 algorithm for the adversarial multi-armed bandit problem to posted price mechanisms, showing a worst-case adversarial bound. Kleinberg and Leighton [15] derive regret bounds for Auer *et al*'s [1] UCB1 bandit algorithm for i.i.d. settings in the posted price context. UCB1 is intended to minimize regret even in finite-horizon contexts, so we would expect it to perform relatively well. However, these algorithms rarely perform very well in terms of utility received in even simulated posted price auction settings – for example, in Conitzer and Garera's comparison of EXP3 with gradient ascent and Bayesian methods [6], or even in different applications, as found by Vermorel and Mohri on an artificially generated dataset and a networking dataset [20]. Conitzer and Garera's Bayesian methods are a relevant comparison to the algorithms we develop here, but they make a "correct prior" assumption, mostly focusing on learning when the model is known but the parameters unknown (for example, when the valuation distribution is uniform or exponential with known probabilities and a set of possible parameters with finite support for each type of distribution).

Contributions.

In this paper, we study the problem of revenue maximization in posted-price auctions of digital goods from the perspective of reinforcement learning and maximizing flow utility, rather than trying to achieve asymptotic regret bounds. We evaluate algorithms on simulated buying populations, with valuations distributed uniformly, exponentially, and log-normally. We find that regret-minimization algorithms from the multi-armed bandit literature are slow to learn in practice, and hence impractical, even for simple distributions of valuations in the buying population. We propose two alternatives: (1) a scheme based on Gittins indices that starts with different priors on the arms based on the knowledge that purchases at higher prices are less likely, and (2) a new reinforcement learning algorithm for the problem, called LLVD, that is based on a plausible linearity assumption on the structure of the demand curve. LLVD maintains a Beta distribution as the seller's belief state, updating it using a moment-matching approximation. LLVD is (approximately) optimal when the linearity assumption holds, and empirically performs well for several families of valuation distributions that violate the linearity assumption.

2. THE POSTED PRICE MODEL

We start by introducing the model and assumptions that we will use. Buyers arrive in a stream, each with an i.i.d. valuation v of the good from an unknown underlying distribution f_V. f_V can have support on $[0, \infty)$, At each instant in time, the seller quotes a price $q_t \in [0, \quad \infty)$, a potential buyer arrives with $v_t \sim f_V$, and chooses to buy if $v_t \geq q_t$ and not to buy otherwise. The seller has access to the history of her own pricing decisions, as well as the purchase decisions made by each arriving buyer. Her goal is to sequentially set q_t so as to maximize (discounted) expected total long-term revenue (we assume an infinite horizon model).

2.1 Learning the Demand Curve

For any given distribution of buyer valuations f_V, under the assumption that buyer valuations are I.I.D. draws from f_V at each point in time, there is a single optimal price q_{OPT} that maximizes the seller's expected revenue. When f_V is unknown, there are several different possible design goals. In this work we seek to design an algorithm that maximizes flow utility, rather than an algorithm with the explicit goal of asymptotically correct or regret-bounded learning. Therefore, we focus on a dynamic programming approach that maximizes flow utility under a probabilistic model. This is a problem that falls within the domain of dynamic programming, reinforcement learning, and optimal experimentation, because the seller's actions, corresponding to posted prices, have both a profit role (exploitation) and an informational role (exploration; conveying information about the true demand curve). The first problem with designing such a model is that the seller's state space is itself a probability distribution over possible probability distributions (of valuations), so without restricting the space of possibilities it is difficult to get any traction. It is useful to consider a simple example.

"Linear" Demand.

Assume that buyer valuations are distributed uniformly on $[0, B]$. The probability of an arriving buyer choosing to buy at price q, $P(q)$ is $(B-q)/B$, or $1-\gamma q$ where $\gamma = 1/B$. This entails a linear form for the probability of a sale at price q, so we refer to this (loosely) as the case of linear demand.

Now consider a particularly simple example. Suppose the seller knows with certainty that the demand function is either F, corresponding to γ_1, or G, corresponding to γ_2. Let α denote the probability the seller associates with demand function F. Then the state space is entirely parameterized by α. The expected discounted revenue is given by $\pi(\alpha_t) = \sum_{k=t}^{\infty} \delta^{k-t}(\alpha_k q_k P_F(q_k) + (1-\alpha_k) q_k P_G(q_k))$.

A revenue maximizing policy is a mapping from α to q that maximizes π. The states $\alpha = 0$ and $\alpha = 1$ have no uncertainty associated with them, and the problem reduces to a simple maximization. When $\alpha = 1$, we maximize $\max_q \sum_{k=0}^{\infty} \delta^k (q P_F(q)) = \max_q \frac{q P_F(q)}{(1-\delta)}$.

For this example we assume $q \in [0, 1]$. So if the optimal q is theoretically greater than 1, the item is priced at 1. The function itself is increasing up to a maximum at $q = 1/2\gamma_1$, so the maximum within our domain $q \in [0, 1]$ is at $q = \min(1/2\gamma_1, 1)$ if $\alpha = 1$. Similarly if $\alpha = 0$, then the optimal price is $q = \min(1/2\gamma_2, 1)$.

For general α, the seller sets a price q (since we are discussing optimal actions in a situation that is not explicitly time dependent, we suppress any dependence on t) Depending on the action of an arriving buyer, the seller updates α. If the buyer buys, then $\alpha^{'} = \frac{\alpha P_F(q)}{\alpha P_F(q) + (1-\alpha) P_G(q)}$. For our particular model, $\alpha^{'} = \frac{\alpha - \gamma_1 \alpha q}{1 + ((\gamma_2 - \gamma_1)\alpha - \gamma_2) q}$. If the buyer does not buy, the state update is $\alpha^{''} = \frac{\alpha(1 - P_F(q))}{\alpha(1-P_F(q)) + (1-\alpha)(1-P_G(q))}$. Again, for our particular model, $\alpha^{''} = \frac{\gamma_1 \alpha}{(\gamma_1 - \gamma_2)\alpha + \gamma_2}$. This latter equation is of particular interest, since there is, surprisingly, no dependence on q.

The relevant probabilities of buying and not buying, given a (state, action) pair consisting of α and q are given by $\Pr(\text{Buy}|\alpha, q) = \alpha P_F(q) + (1-\alpha) P_G(q)$, and $\Pr(\neg\text{Buy}|\alpha, q) = \alpha(1 - P_F(q)) + (1-\alpha)(1 - P_G(q))$.

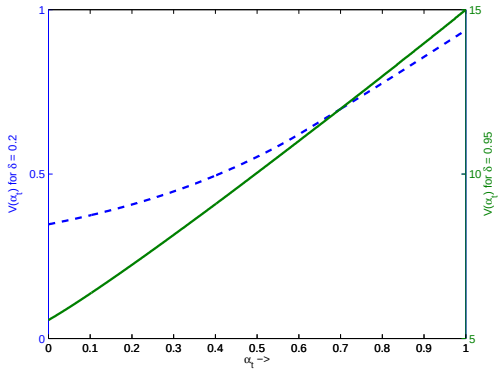


Figure 1: The value function for $\gamma_1 = 0.25, \gamma_2 = 0.9$ and discount factors $\delta = 0.2$ and $\delta = 0.95$. Note how the value function for high δ is almost linear.

Now we can write down the Bellman equation:

$$V(\alpha) = \alpha q P_F(q) + (1-\alpha) q P_G(q) + \delta V' \tag{1}$$

where $V' = \Pr(\text{Buy}|\alpha, q)V(\alpha^{'}) + \Pr(\neg\text{Buy}|\alpha, q)V(\alpha^{''})$.

We now know the dynamics of the system. We can solve by discretizing α (we know $\alpha \in [0,1]$) and using value iteration for any particular values of γ_1 and γ_2. Figure 1 shows the value function for two different values of δ.

Computing the value function in this case leads to an interesting observation. When δ is high, the value function is almost linear in α. We can approximate the value function by $V = b\alpha + c$ to get an analytical approximate solution. Substituting in Equation 1 and finding b and c by equating coefficients, we find $V = \frac{Zq^2+q}{1-\delta}$ where $Z = (\gamma_2 - \gamma_1)\alpha - \gamma_2$. This equation implies that the optimal choice for q is the same as the myopically optimal choice! The linearity of the value function and the approximate optimality of a myopic strategy arise in part because, regardless of the strategy for setting q, good information is received by whether or not a buyer buys, allowing us to distinguish the populations, and α converges to either 0 or 1 quickly. This is partly a function of the fact that only one of the two possible future states α' and α'' depends in any way on q. In fact, the myopic approximation continues to be an excellent approximation to the optimal strategy even for lower values of δ, because at lower values immediate revenue dominates future revenue in the value function anyhow.

More General Settings.

The example discussed above is analytically tractable because of the restriction to two possible distributions, reducing our state space to a single continuous variable. This restriction is too onerous for any realistic application. The simplest way to remove this restriction without sending tractability overboard is to consider the whole space of linear demand functions with $\gamma \in [0,1]$ (the restriction to $\gamma \leq 1$ is not restrictive, because the effect could be achieved through rescaling of the valuations). We approach this problem by maintaining a probability distribution over γ.

3. ALGORITHMS

Here we describe the three algorithms we compare for this problem: (1) our new parametric algorithm, LLVD; (2) a Gittins-index based strategy with appropriately chosen priors; (3) UCB, a regret-minimizing algorithm from the multi-armed bandit literature.

3.1 The LLVD Algorithm

Our main assumption is that it is reasonable to model the probability of an arriving buyer choosing to go through with a purchase at quoted price q as a linear function of q, $\Pr(\text{Buy}|q) = 1 - \gamma q$. This gives rise to our learning algorithm, which we call "Linear Learning of Valuation Distributions" (LLVD).

Under the linearity assumption we want to maximize total expected (discounted) revenue. The seller's state space is now the space of distributions over γ. In order to make this a tractable state space to work with, we enforce that the seller always represents her beliefs as a Beta distribution ($\gamma \in [0,1]$). The state space can then be parametrized by the two parameters of the Beta distribution. We need to derive the state space transition model and the reward model in order to solve for the seller's optimal policy. In the following, $f(\gamma; \alpha, \beta)$ represents the density function for the Beta distribution. $F(\gamma; \alpha, \beta)$ represents the c.d.f for the Beta distribution, and $F_k(\gamma)$ represents $F(\gamma; \alpha + k, \beta)$.

Transition Model.

An arriving buyer is quoted a price q and decides whether or not to buy at that price. She will buy if her valuation is less than equal to the price quoted. The seller updates her own distribution over γ based on whether or not the arriving buyer bought the good. Consider the Bayesian updates in two cases:

1. Buyer does not buy:

$$f(\gamma|\neg\text{Buy}) = \frac{f(\gamma;\alpha,\beta)(\gamma q)}{\int_0^{1/q} f(\gamma;\alpha,\beta)(\gamma q)d\gamma} = \frac{\gamma^{\alpha}(1-\gamma)^{\beta-1}}{\int_0^{1/q}\gamma^{\alpha}(1-\gamma)^{\beta-1}d\gamma}$$
$$= \frac{f(\gamma;\alpha+1,\beta)}{F(1/q,\alpha,\beta)} = \frac{f(\gamma;\alpha+1,\beta)}{F_0(1/q)}$$

For $q < 1$, the normalizing constant is 1 and the true posterior is Beta. When $q > 1$ the posterior need not be Beta, so we compute the Beta distribution that matches the first and second moment of the true posterior. This yields a pair of simultaneous equations for α_{t+1} and β_{t+1} (in the equations below F_k represents $F_k(1/q_t)$):

$$\frac{\alpha_{t+1}}{\alpha_{t+1}+\beta_{t+1}} = \frac{q_t\mathbb{E}(\gamma^2)F_2 + \mathbb{E}(\gamma)(1-F_1)}{(q_t\mathbb{E}(\gamma)F_1 + 1 - F_0)}$$
$$\frac{\alpha_{t+1}(\alpha_{t+1}+1)}{(\alpha_{t+1}+\beta_{t+1})(\alpha_{t+1}+\beta_{t+1}+1)} = \frac{q_t\mathbb{E}(\gamma^3)F_3 + \mathbb{E}(\gamma^2)(1-F_2)}{(q_t\mathbb{E}(\gamma)F_1 + 1 - F_0)}$$

2. Buyer buys:

$$f(\gamma|\text{Buy}) = \frac{f(\gamma;\alpha,\beta)(1-\gamma q)}{\int_0^{1/q} f(\gamma;\alpha,\beta)(1-\gamma q)d\gamma}$$
$$= \frac{f(\gamma;\alpha,\beta)(1-\gamma q)}{(F(1/q,\alpha,\beta) - q\mathbb{E}(\gamma)F(1/q,\alpha+1,\beta))}$$
$$= \frac{f(\gamma;\alpha,\beta)(1-\gamma q)}{(F_0(1/q) - q\mathbb{E}(\gamma)F_1(1/q))}$$

Again, we approximate the true posterior with a Beta distribution by matching the first and second moments.

$$\frac{\alpha_{t+1}}{\alpha_{t+1}+\beta_{t+1}} = \frac{\mathbb{E}(\gamma)F_1 - q_t F_2 \mathbb{E}(\gamma^2)}{F_0 - q_t \mathbb{E}(\gamma)F_1}$$

$$\frac{\alpha_{t+1}(\alpha_{t+1}+1)}{(\alpha_{t+1}+\beta_{t+1})(\alpha_{t+1}+\beta_{t+1}+1)} = \frac{\mathbb{E}(\gamma^2)F_2 - q_t \mathbb{E}(\gamma^3)F_3}{F_0 - q_t \mathbb{E}(\gamma)F_1}$$

Let M and S represent first and second order moments respectively. Solving these equations yields update rules $\alpha_{t+1} = \frac{MS-M^2}{M^2-S}$ and $\beta_{t+1} = \frac{(1-M)\alpha_{t+1}}{M}$.

Reward Model.

Let π denote the discounted long-term revenue and δ the discount factor. Let $P(q) = \Pr(\text{Buy}|q)$. Then $\pi = q_0 P(q_0) + \sum_{t=1}^{\infty} q_t P(q_t)$. The first term, $\pi_0 = q_0 P(q_0)$ is the expected reward at this particular instant, from the next action. We can compute the expected value of this term:

$$\begin{aligned} P(q) &= \int_0^{1/q} (1-\gamma q) f(\gamma;\alpha,\beta) d\gamma \\ &= F(1/q;\alpha,\beta) - q\mathbb{E}(\gamma)F(1/q;\alpha+1,\beta) \\ &= F(1/q;\alpha,\beta) - q\mu F(1/q,\alpha+1,\beta) \end{aligned} \quad (2)$$

where $\mu = \alpha/(\alpha+\beta)$.

$$\begin{aligned} \pi_0 &= q_0(F(1/q_0;\alpha,\beta) - q_0\mathbb{E}(\gamma)F(1/q_0;\alpha+1,\beta)) \\ &= q_0(F(1/q_0;\alpha,\beta) - q_0\mu F(1/q_0,\alpha+1,\beta)) \end{aligned} \quad (3)$$

The Bellman Equation.

In a risk-neutral framework, we can similarly take expectations over γ and derive the appropriate Bellman equation: $V(\alpha_t,\beta_t) = \max_q qP(q) + \delta V'$, where

$$V' = P(q)V(\alpha_{t+1},\beta_{t+1}|\text{Buy}) + (1-P(q))V(\alpha_{t+1},\beta_{t+1}|\neg\text{Buy})$$

Obviously, if γ were known to the seller, the optimal action would be the optimal myopic action, and it would yield a discounted expected revenue of:

$$\begin{aligned} \pi &= \max_q (q(1-\gamma q) + \sum_{t=1}^{\infty} \delta^t q(1-\gamma q)) \\ &= \max_q \frac{q(1-\gamma q)}{1-\delta} = \max_q \frac{q(1-\gamma q)}{1-\delta} \end{aligned} \quad (4)$$

This equation is maximized at $q = \frac{1}{2\gamma}$, in our environment, yielding $V = \frac{1}{4\gamma(1-\delta)}$.

Solving for the optimal policy.

Various issues arise in trying to solve such a system. A value-iteration type method would rely on a reasonable functional approximation of the value function in order to converge to a correct estimate. We use a different approach by first restricting the problem to a space where table-based value iteration can be applied, and then extrapolating to the complete space. We start by restricting to values of q between 0 and 1.

The $q < 1$ case: Equation 2 reduces to $P(q) = (1-\mu q)$, therefore Equation 3 reduces to $\pi_0 = q_0(1-\mu q_0)$ because $F(1/q) = 1$ for the Beta distribution as $q < 1$. Equation 4 is maximized at $q = \min(1, \frac{1}{2\mu})$, in our environment, yielding $V = \min(\frac{1}{4\mu(1-\delta)}, \frac{1-\mu}{1-\delta})$. Since the transition model is known

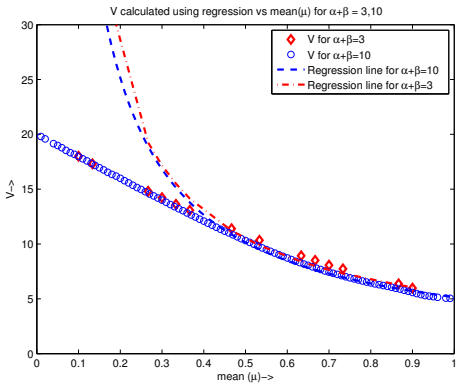


Figure 2: Comparison of the regression line with data from the value iteration table for different values of $\alpha+\beta$. Note the very tight match in the domain where the optimal q would be expected to be less than 1. The regression function allows LLVD to generalize this to the entire space (notice the difference between the line and the data points for lower values of μ, which correspond to higher optimal values of q).

(the fact that the true posterior is Beta when the buyer buys for $q < 1$ is helpful in efficient implementation), all that remains in order to discretize and apply value iteration is to specify some boundary conditions on the model. The boundary conditions correspond to having a high degree of certainty about the value of γ. We assume that when the variance of the Beta distribution becomes less than 0.001, γ can be assumed to be known to the seller, and it is then equal to μ. In order for this technique to be consistent, we need to show that once the variance is sufficiently low, it will not be the case that it again starts increasing. We can show that in expectation the variance decreases in every iteration for $q < 1$; the proof is omitted due to space considerations.

This yields the final algorithm: we use value iteration to solve for the value function on a grid for $\alpha, \beta \in [0.1, 200]$, but we pre-fill all spaces where α, β are such that the variance of the distribution is less than 0.001. Figure 3 (V1) shows the value function for $\delta = 0.95$, as a function of α and β.

Extending to $q > 1$: We expect the value function computed using table-based value iteration to closely approximate the universally "correct" one for regions where the optimal value of q is less than 1. Therefore, we fit a regression line using values from the value function matrix where $\mu > 0.6$ (implying that the optimal q is probably lower than 0.85). Empirically, we find that the value function is close to linear in $\frac{1}{\mu}$ and $\frac{1}{\alpha+\beta}$ (see Figure 2). So we approximate the value function for the whole space as

$$V(\alpha,\beta) = a_1 \frac{\alpha+\beta}{\alpha} + a_2 \frac{1}{\alpha+\beta} \quad (5)$$

Figure 3 shows that this is a good approximation over the entire space. Now, at any time $T = t$, with the belief state

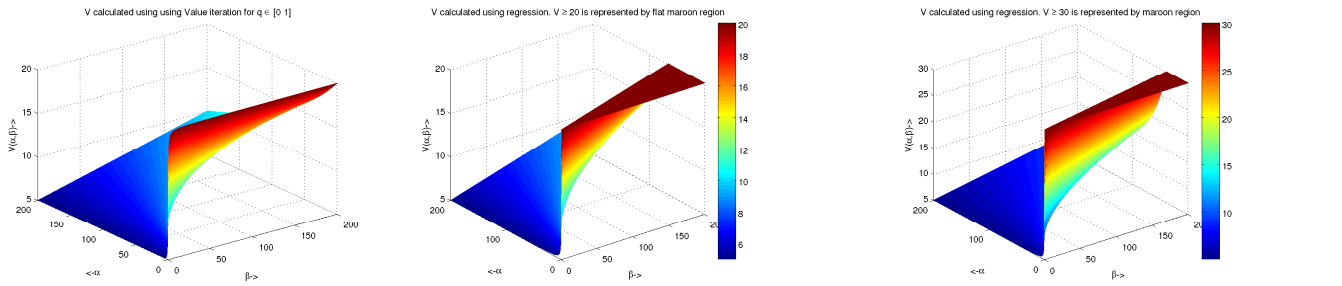


Figure 3: V1 is the value function computed using table-based value iteration with $q < 1$ (the maximum value of V is 20). V2 is the value function computed using regression (see Equation 5), showing the similarity to V1 where the value function is less than 20 (the flat maroon region shows where $V \geq 20$, where the value functions would be expected to differ, and $q > 1$). V3 shows some more of the structure of the value function computed using regression (Equation 5) in the region where it attains values between 20 and 30.

(α, β) we can find the q which maximizes the given equation.

$$\pi = \max_{q_t} V(\alpha, \beta) + \delta(\Pr(\text{Buy}|q_t)V(\alpha', \beta'|\text{Buy}) + (1 - \Pr(\text{Buy}|q_t))V(\alpha'', \beta''|\neg\text{Buy}))$$

Here α', β', α'' and β'' are functions of q_t,α and β, price offered at time T=t. These values can be calculated as discussed above by comparing the first two moments.

Implementation notes: In our experiments, we compute the value function using $\delta = 0.95$. The best fit regression line is obtained for $a_1 = 4.99$ and $a_2 = 1.5147$; for convenience we use $a_1 = 5$ and $a_2 = 1.5$. The LLVD based seller then learns online, constantly updating her belief on γ (starting from $\alpha = \beta = 1$), and choosing the price that maximizes the value function at any instant.

3.2 Bandit Schemes

Multi-armed bandit algorithms are often applied to Dynamic pricing [16]. The different pricing options are the arms of the bandit and the goal is to find the arm that maximizes infinite horizon discounted reward. The downside of such approaches is that one needs to have fixed arms, and there is no "information sharing" between arms. How to discretize the space into arms is an interesting problem. For the purposes of this paper, we discretize the space from [0.5 $2q^*$] in 20 steps, where q^* is the (analytically computed) optimal price for the specific valuation distribution. While reasonable for evaluation, there may be situations where the need to find a reasonable interval is a downside for bandit-based methods. We discuss two algorithms.

A Gittins Index Scheme With Smart Priors.

Gittins and Jones introduced dynamic allocation indices as the Bayes optimal solution to the exploration-exploitation dilemma in the standard multi-armed bandit context [10, 8, 9]. In the context of "yes/no" rewards, a particularly useful, computable scheme is to maintain a Beta prior on each arm. This takes advantage of the conjugate nature of the Beta distribution for Bernoulli observations. The distribution $\beta(a, b)$ is updated to $\beta(a + 1, b)$ upon success and $\beta(a, b + 1)$ upon failure. For every pair (a, b) we can calculate the Gittins index $G(a, b)$. For simplicity we assume that when $a+b \geq 500$, the mean $\frac{a}{a+b}$ represents the correct probability of success for that arm. We choose the arm to play next by multiplying the Gittins index for each arm with its payoff if the arm is successful, $S_i = q_i G(a_i, b_i)$ and choosing the arm with highest S_i. This is equivalent to maintaining Gittins indices on arms with two payoffs, 0 and q_i [16].

Parameters: Price $Q \in [0.5, 2q^*]^K$, Matrix G of Gittins Indices.
Initialization: $n = 0$ (# buyers so far) , Divide Q in 4 regions in increasing order of magnitude. Initialize state S for each of the K arms according to the region they lie in: from lower to higher: (4,1),(3,2),(2,3),(1,4)

For each k in Buyers do:

1. Price the item at Q_j which maximizes $Q_j.G[S_j]$. Denote the chosen price by Q_{j*}.
2. If the buyer buys, set $S_j(a) = S_j(a) + 1$ else set $S_j(b) = S_j(b) + 1$

Table 1: A Gittins-Index Based Algorithm. The K parameter governs the discretization of the space (we use $K = 20$).

The standard approach of initializing all the arms with the same prior is inappropriate in this case, because we know that the probability of a buyer buying at a higher price is lower. Thus we arrange the arms in increasing order of their weights and divide them in 4 region. We initialize arms in the region with lowest weight with a Beta $(4, 1)$ prior, the next lowest with a Beta $(3, 2)$ prior, next with $(2, 3)$ and the remaining with $(1, 4)$. As expected, this weighting of the priors significantly outperforms uniform priors on all the arms. Table 1 shows the final algorithm in detail.

UCB1.

Much work on digital goods auctions has focused on algorithms with good regret bounds. Two of these that are based on algorithms for multi-armed bandit problems have gained particular attention, namely the EXP3 algorithm [2, 3] and the UCB1 algorithm [1, 15]. Kleinberg discusses a "continuum armed" bandit algorithm called CAB1, which is

Parameters: Price $Q \in [0.5, 2q^*]^K$, Number of buyers: *nob*.
Initialization: $n = 0$ (# buyers so far)

For each k in first K buyers do:

1. Price the item at Q_k
2. $n_k = 1; n = n + 1$
3. If the buyer buys then $x_k = Q_k$ else $x_k = 0$

For the remaining buyers at each time instant t do:

1. Price the item at Q_j which maximizes $\frac{x_j}{n_j} + \sqrt{\frac{2 \ln n}{n_j}}$. Denote the chosen price by Q_{j*}.
2. $n_{j*} = n_{j*} + 1; n = n + 1$
3. If the buyer buys, set $x_{j*} = x_{j*} + Q_{j*}$ and update total profit

Table 2: Algorithm UCB1, adapted to our setting. The K parameter governs the discretization of the space (we use $K = 20$).

a wrapper around algorithms like UCB1 or EXP3 for continuous spaces [14]. We perform extensive empirical tests on all these algorithms, adapted to our setting. UCB1 and EXP3 discretize the action space and treat each possible price as a unique possible action (or "arm" in bandit language). The EXP3 and UCB1 algorithms are specifically designed for adversarial and I.I.D scenarios respectively. As expected, we find that EXP3 is outperformed (or equaled in performance) by UCB1 in all our I.I.D. scenarios, so we do not report results from EXP3. While one would expect CAB1 to perform well, since it is designed for continuous action spaces, it is geared more towards producing useful regret bounds, and does not take advantage of the structure of the search space, instead using doubling processes to efficiently scan a potentially large continuum. It is outperformed by UCB1. The specific form of the UCB1 algorithm we use is shown in Table 2.

4. EXPERIMENTAL RESULTS

We consider various different distributions that generate demand. We restrict ourselves to I.I.D. assumptions rather than considering adversarial scenarios.

Choice of distributions.

We consider three sets of valuation distributions that generate a wide range of optimal prices:

1. Uniform on $[0, B]$ where B is $4, 2.5, 1.5$.
2. Exponential with rate (λ) parameters $1.75, 0.8, 0.5$.
3. Log-normal with location (μ) and scale (σ) parameters $(1, 1)$, $(1, 0.75)$ and $(1, 0.5)$.

Analysis of Results.

Each simulation consists of a stream of n buyers, arriving one after the other, each buyer has a valuation v that is sampled at random from the valuation distribution. The seller chooses a price q to offer, and if $v \geq q$ the buyer goes through with the purchase, otherwise she turns down the offer. In Figure 4 we report results averaged over 1000 simulations of the process, each consisting of 500 time steps.

In addition to comparing the algorithms, in cases where the linearity assumption of LLVD is violated (exponential and log-normal valuation distributions), we are interested in quantifying how much of the regret of the algorithm can be attributed to the linearity assumption itself, and how much may be due to not learning the best possible linear function. In order to study this, we also report the analytical profit that would be achieved by using the linear function of the form $1 - \gamma q$ to model the probability of buying, when γ is chosen so that the functional distance between the uniform distribution on $[0, 1/\gamma]$ and the true target valuation distribution is minimized. We evaluate functional distance between the two distributions as the sum of squared difference between their c.d.f (square of L2-Norm of the difference of the c.d.f). Let $F(x)$ and $G(x)$ be the two distributions

$$f_d = \text{ L2-Norm } = \sqrt{\int_0^\infty (F(x) - G(x))^2 \, dx}$$

In our case where $F(x)$ is the uniform distribution in the interval $[0, B]$ where $B = 1/\gamma$.

$$D = {f_d}^2 = \int_0^B (F(x) - G(x))^2 \, dx + \int_B^\infty G^2(x) \, dx$$

Further details are in Appendix A.

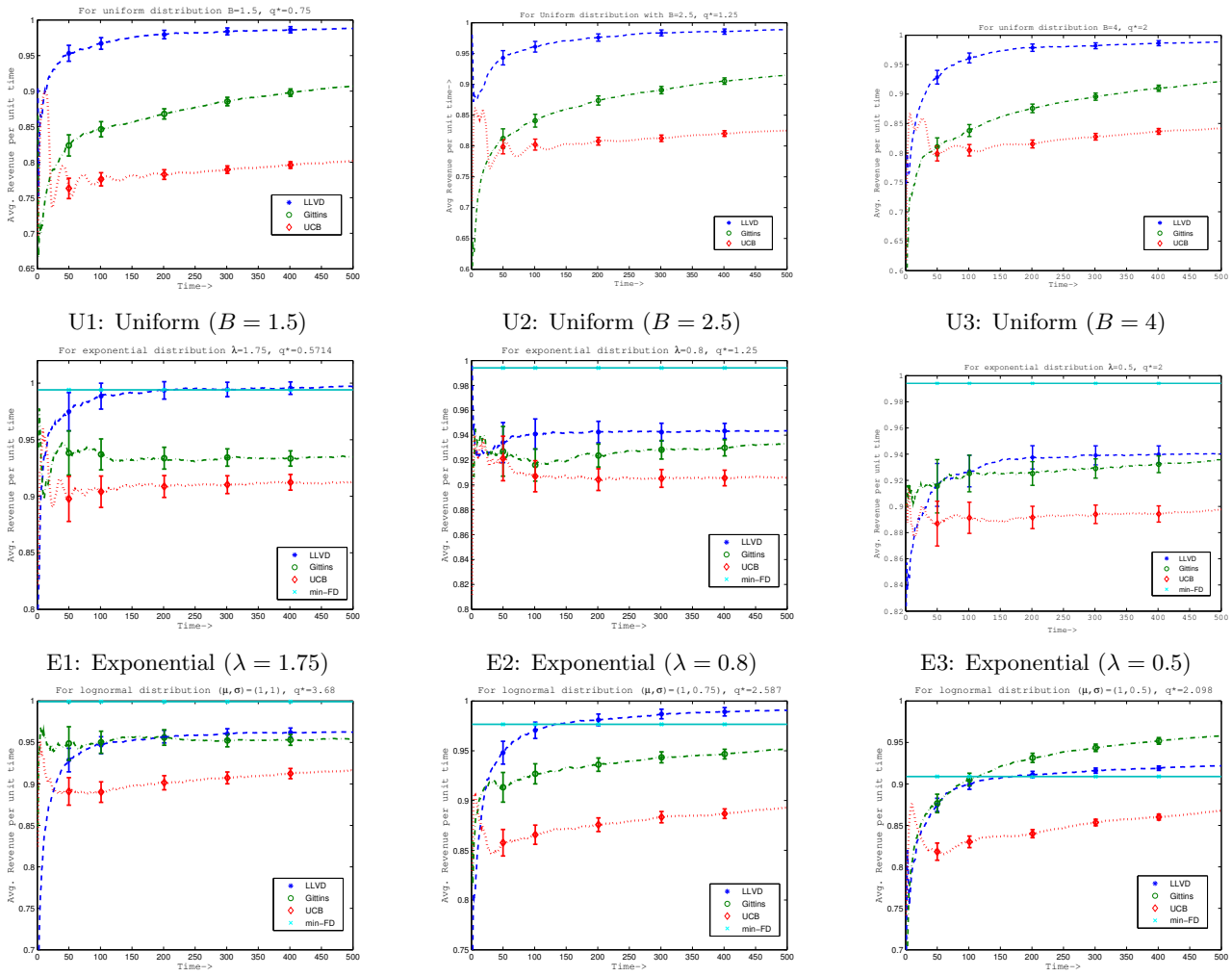
Uniform valuation distributions (linear demand) As expected, LLVD always learns the correct distribution rapidly in these cases, significantly outperforming UCB1 and the Gittins-index based scheme.

Exponential valuation distributions In this case, $\Pr(\text{Buyer Buys}|q) = e^{-\lambda q}$, where λ is the rate parameter. LLVD performs either better than or as well as the Gittins-index based scheme in these cases, and significantly outperforms UCB1.

Log-normal valuation distributions For the log-normal, $\Pr(\text{Buyer Buys}|q) = 1 - \phi(\frac{\ln q - \mu}{\sigma})$, where μ and σ are the location and scale parameters for the log-normal distribution. While LLVD dominates UCB1, the Gittins-index based scheme is competitive, sometimes performing better and sometimes worse. LLVD may have trouble with these cases because the log-normal distribution is harder to approximate with a linear function, or because the learning process is thrown off. In some cases LLVD even outperforms the "best" linear function (indicating that the fit over the entire distribution is not necessarily the best measure when profit-seeking behavior is determined by only a portion of the distribution), providing evidence for the latter explanation.

A note about long-term learning.

It is worth noting that in the long-term, when the LLVD algorithm converges to a suboptimal price, it remains suboptimal, whereas bandit-based algorithms keep learning and slowly improving their performance over time. In some cases (like exponential distributions with $\lambda = 0.5, 0.8$) where LLVD and the Gittins index scheme perform similarly, the perfor-



U1: Uniform ($B = 1.5$) U2: Uniform ($B = 2.5$) U3: Uniform ($B = 4$)

E1: Exponential ($\lambda = 1.75$) E2: Exponential ($\lambda = 0.8$) E3: Exponential ($\lambda = 0.5$)

L1: Log-normal ($\mu = 1, \sigma = 1$) L2: Log-normal ($\mu = 1, \sigma = 0.75$) L3: Log-normal ($\mu = 1, \sigma = 0.5$)

Figure 4: Main experimental results: Each graph shows the time-averaged profit received at any time, averaged over 1000 simulations and the 95% confidence interval. The top row shows uniform valuation distributions, corresponding to the model LLVD is based on. The second row shows exponential valuation distributions, and the bottom row log-normal ones. All values are represented as fraction of optimal profit.

mance of the Gittins index scheme continues to improve over time, eventually exceeding that of LLVD. Our primary interest is in maximizing revenue in the initial stages, because we assume that over time the distribution can be learned anyhow, perhaps in an "off-policy" manner.

5. DISCUSSION

As dynamic pricing becomes a reality with intelligent agents making rapid pricing decisions on the Internet, the field of algorithmic pricing has developed rapidly. While there has been continuing work on revenue management and inventory issues in operations research, the study of posted price mechanisms for digital goods auctions has mostly been confined to theoretical computer science, inspired by developments from computational learning theory. As a result, the focus has mostly been on deriving regret bounds rather than developing and analyzing algorithms that could prove useful in practice. In the spirit of Vermorel and Mohri's empirical analysis of algorithms for bandit problems [20], we believe that it is important to test algorithms in simulation, and ideally in real-world environments, or at least using real-world data. This paper starts exploring this path with simulation experiments.

We find that the UCB1 algorithm, which has some desirable theoretical properties for posted price auctions with unlimited supply, can be slow to learn in simple simulated environments; further, choosing the right number of arms can have a significant effect on performance (we experimented with several different numbers of arms to come up with a good number, reported in this paper). Theoretical extensions to spaces with a continuum of actions, like CAB1, fare no better. However, there are two promising directions: (1) an algorithm based on making a linearity assumption about the demand curve performs well, even when the true model

is not linear. Additionally, our experimental results and theoretical analysis of the linearity assumption indicate that it may be a very useful approximation, far beyond just for truly linear models. (2) Using simple but appropriate priors in a Gittins-index based scheme also shows promise. There is still scope to further improve performance by enabling better information sharing between arms. One possibility is to apply knowledge gradient techniques [18, 17] to the pricing problem, but current state-of-the-art KG techniques also do not account for correlation between arms. Existing extensions typically consider multivariate normal priors, though, which are not appropriate for monotonic functions like demand. This is a fruitful area for future work.

6. ACKNOWLEDGMENTS

We are grateful for research funding from an NSF CAREER award (0952918), and from a US-Israel BSF Grant (2008404). We thank David Sarne and Malik Magdon-Ismail for several helpful conversations.

7. REFERENCES

[1] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2):235–256, 2002.

[2] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Proc. FOCS*, volume 36, pages 322–331. IEEE Computer Society Press, 1995.

[3] A. Blum, V. Kumar, A. Rudra, and F. Wu. Online learning in online auctions. *Theoretical Computer Science*, 324(2-3):137–146, 2004.

[4] T. Chakraborty, Z. Huang, and S. Khanna. Dynamic and non-uniform pricing strategies for revenue maximization. In *Proc. FOCS*, 2009.

[5] Y. Chen and R. Wang. Learning buyers' valuation distribution in posted-price selling. *Economic Theory*, 14(2):417–428, 1999.

[6] V. Conitzer and N. Garera. Learning algorithms for online principal-agent problems (and selling goods online). In *Proceedings of the 23rd international conference on Machine learning*, pages 209–216. ACM, 2006.

[7] S. Das and M. Magdon-Ismail. Adapting to a market shock: Optimal sequential market-making. In *Advances in Neural Information Processing Systems (NIPS)*, pages 361–368, 2008.

[8] J. Gittins. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society. Series B (Methodological)*, 41(2):148–177, 1979.

[9] J. Gittins. *Multi-armed bandit allocation indices*. John Wiley & Sons Inc, 1989.

[10] J. Gittins and D. Jones. A dynamic allocation index for the discounted multiarmed bandit problem. *Biometrika*, 66(3):561–565, 1979.

[11] A. Goldberg and J. Hartline. Envy-free auctions for digital goods. In *Proc. ACM EC*, pages 29–35. ACM New York, NY, USA, 2003.

[12] A. Goldberg, J. Hartline, and A. Wright. Competitive auctions for multiple digital goods. In *Proc. ESA*, pages 416–427. Springer, 2001.

[13] J. Kephart, J. Hanson, and A. Greenwald. Dynamic pricing by software agents. *Computer Networks*, 32(6):731–752, 2000.

[14] R. Kleinberg. Nearly tight bounds for the continuum-armed bandit problem. *Advances in Neural Information Processing Systems*, 18, 2005.

[15] R. Kleinberg and T. Leighton. The value of knowing a demand curve: Bounds on regret for on-line posted-price auctions. In *Proc. FOCS*, 2003.

[16] M. Rothschild. A two-armed bandit theory of market pricing. *Journal of Economic Theory*, 9(2):185–202, 1974.

[17] I. Ryzhov, P. Frazier, and W. Powell. On the robustness of a one-period look-ahead policy in multi-armed bandit problems. *Procedia Computer Science*, 1(1):1629–1638, 2010.

[18] I. Ryzhov, W. Powell, and P. Frazier. The knowledge gradient algorithm for a general class of online learning problems. *Submitted for publication*, 2008.

[19] I. Segal. Optimal pricing mechanisms with unknown demand. *American Economic Review*, 93(3):509–529, 2003.

[20] J. Vermorel and M. Mohri. Multi-armed bandit algorithms and empirical evaluation. In *Proc. ECML*, pages 437–446. Springer, 2005.

APPENDIX

A. FUNCTIONAL DISTANCE

Let $F(x) = \frac{x}{B}$ represent the c.d.f of Uniform distribution over the interval [0 B] and $G(x)$ be the c.d.f be the actual valuation distribution. L2-Norm for the difference between the two distributions is given by:

$$f_d = \sqrt{\int_0^\infty (F(x) - G(x))^2 \, dx}$$

For convenience we consider $D = {f_d}^2$, written as

$$D = {f_d}^2 = \int_0^\infty ((1 - G(x)) - (1 - F(x)))^2 \, dx$$

Let $F_1(x) = 1 - F(x)$ and $G_1(x) = 1 - G(x)$. Then

$$\begin{aligned} D &= \int_0^B F_1^2(x) \, dx - 2\int_0^B G_1(x)F_1(x) \, dx + \int_0^\infty G_1^2(x) \, dx \\ &= \frac{B}{3} - 2\int_0^B G_1(x)F_1(x) \, dx + \int_0^\infty G_1^2(x) \, dx \end{aligned}$$

differentiating w.r.t B and setting to 0 to calculate minima, we find

$$\frac{1}{3} - 2\int_0^B \frac{qG_1(x)}{B^2} \, dx = 0$$

This equation can easily be solved numerically for $G(x)$ exponential and lognormal respectively, and it can be verified that $\frac{d^2D}{dB^2} > 0$ for minima.