Algorithmic pricing with strategic consumers

Andreas Bech

USC

January 21, 2026

Abstract. Abstract

Keywords. Pricing, monopolist, reinforcement learning.

Contact. Emails. Thanks to.....

1. INTRODUCTION

Machine learning/AI is widespread in business operations, pricing, recommendation, fraud detection, etc. In this paper I on focus algorithmic pricing using so-called learning algorithms.

The problem is that firms are using algorithmic pricing but rational consumers might anticipate prices and adapt to algorithms. Little is known about how common algorithms, which are single-player algorithms, optimized on an exogenous environment, perform with strategic adversaries. An interaction between algorithm and best responding consumer is potentially complex and untractable.

2. LITERATURE

- Machine Learning and Dynamic Pricing: Den Boer (2015), Misra, Schwartz and Abernethy (2019)
- Algorithmic Collusion: Calvano et al. (2020), Hansen, Misra and Pai (2021), Klein (2021)
- Strategic consumers: Su (2007)

3. MODEL

The game proceeds over $t = 1, 2, \ldots, T$, where a monopolists sets a price p_t each period in order to maximize average profit. In each period a new consumer is drawn according to $v_t \sim F$ and lives for at most two periods. If the consumer buys in the first period they exit, otherwise they are alive in the second period.

The consumer is assumed to discount the future by β, and given a belief about the future price, p^e_{t+1}, the consumer buys in the first period if

$$v_t - p_t \geq \max\{0, \beta(v_t - p^e_{t+1})\}$$

F is assumed to be unknown to the monopolist such that it uses a pricing policy Ψ that learns from past experience: $p_t = \Psi\left(\{p_\tau, \pi_\tau \mid \tau = 1, \ldots, t-1\}\right)$. The consumer forms beliefs about the future price based on past prices.

3.1. Regret

Let the prices the firm can choose be from a grid of size K. The regret of a policy, Ψ, through time t is

$$
\begin{aligned}
\text{Regret}\,(\Psi, t) &= \mathbb{E}\left[\sum_{\tau=1}^{\mathrm{t}} \pi^* - \pi_{p_\tau}\right] \\
&= \sum_{\tau=1}^{t} \left(\pi^* - p_\tau(1 - F(p_\tau))\right) \\
&= \pi^* t - \sum_{k=1}^{K} p_k(1 - F(p_k))\mathbb{E}\left[n_{kt}\right]
\end{aligned}
$$

It is the ex-post regret from not choosing the constant, optimal price p^*

3.2. Multi-Armed Bandits and the UCB1 Algorithm

The so-called multi-armed bandits problem is a sequential decision-making problem where an agents has to choose one of multiple options (arms) in each period. Each arm yields a reward with an unknown distribution. The goal to maximize cumulative reward. When choosing among the arms the agent faces the famous exploration vs. exploitation tradeoff. One of the most common algorithms for balancing this tradeoff is the **UCB1** algorithm: At time t select arm (price) k that maximizes:

$$
\bar{\pi}_k + \sqrt{\frac{2 \ln t}{n_{kt}}}
$$

where $\bar{\pi}_k$: average profit for arm k, t: total plays, n_{kt}: plays for arm k at time t.

3.3. Q-learning

Another framework for sequential learning is reinforcement learning. Reinforcement learning assumes the agent faces an unknown Markov decision process where rewards in each period depends on the state, s, and the action, a. One of the simpler, and very popular, algorithms in this framework is Q-learning. Q-learning aims to learn a value function $Q(s, a)$ through interacting with the Markov process. It is model-free, meaning that it does not know the transition probabilities.

The goal is again to maximize the expected cumulative reward and this is done by estimating the expected value of taking action a in state s, $Q(s, a) = \mathbb{E}[R_t + \gamma \max_{a'} Q(s', a')]$. This is done by learning from experience using the update rule

$$
Q(s, a) \leftarrow Q(s, a) + \alpha \left(r + \gamma \max_{a'} Q(s', a') - Q(s, a)\right)
$$

, where α is the learning rate, γ is the discount factor, r is the reward, and s' is next state.

4. BENCHMARK REGRET BOUND

4.1. Single period consumers

When consumer only live one period (equivalently $\beta = 0$) it is a well known case in the multi-armed bandit literature. The main result from that literature is that the Regret grows logarithmically (sublinearly)

$$\text{Regret}\,(\Psi, t) = \mathcal{O}(\log T).$$

See Auer et al. (2002).

4.2. Two period game

In a simplified version of the game where $T = 2$ and only a single consumer, the solution can be formulated in a game theory framework. Assume that it is a two period game with no commitment and that F is Uniform(0, 1). In this case the equilibrium price is

$$p_1 = \frac{(2-\beta)^2}{2(4-3\beta)} < 1/2$$

If the consumer doesn't buy in the first period the second period price is

$$p_2 = \frac{(2-\beta)}{2(4-3\beta)} < 1/2$$

Minimum value for either is 0.45.

4.3. Two period game with two consumers

A slightly extended version of the previous game is when $T = 2$ but a second consumer enters in the second period. The monopolist has to set same price p_2 for both old and new consumer, if old consumer does not buy in period 1. This model does not have a closed form solution. The plot below shows the equilibrium prices as a function of the discount factor.

If the monopolist discounts but consumer does not the solution is slightly different, as seen in figure 2.

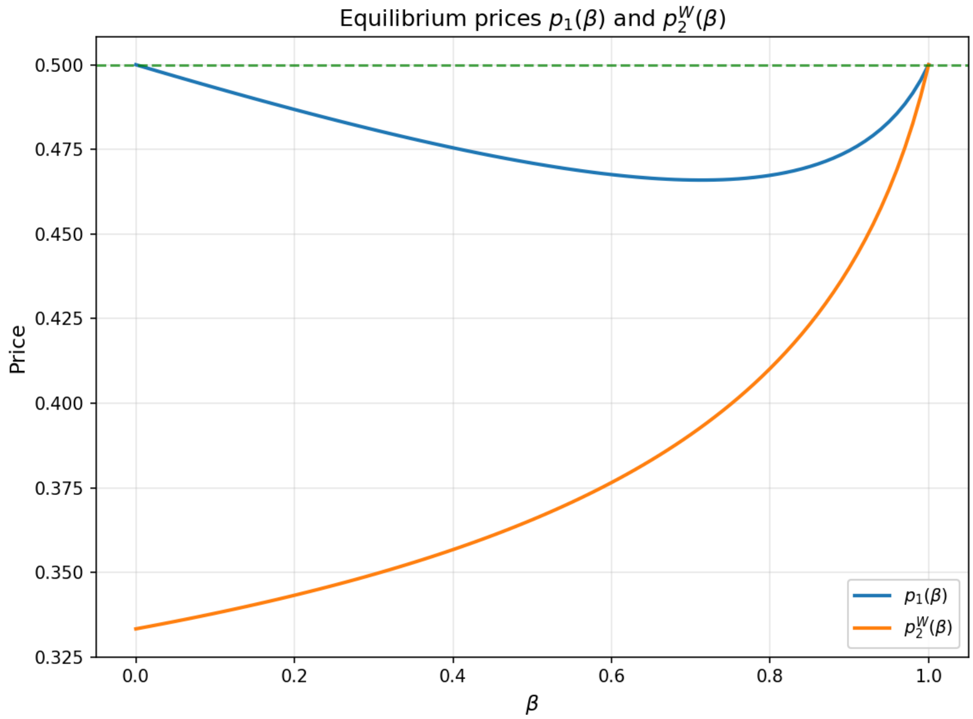

Figure 1: Two period game with two consumers

5. SIMULATIONS

5.1. UCB1

F is assumed to be Uniform(0,1). Monopolist sets price using $p_t = \textbf{UCB1}\left(\{p_\tau, \pi_\tau \mid \tau = 1, \ldots, t-1\}\right)$. Consumer at time t expects future price $p_{t+1}^e = \bar{p}_t$.

5.2. Thompson

Thompson sampling is alternative MAB algorithm.

5.3. Q-learning

The state is whether previous consumer is still alive, along with the previous price (20 states)

6. EXTENSIONS

7. DISCUSSION

- Find a proper nested stage game

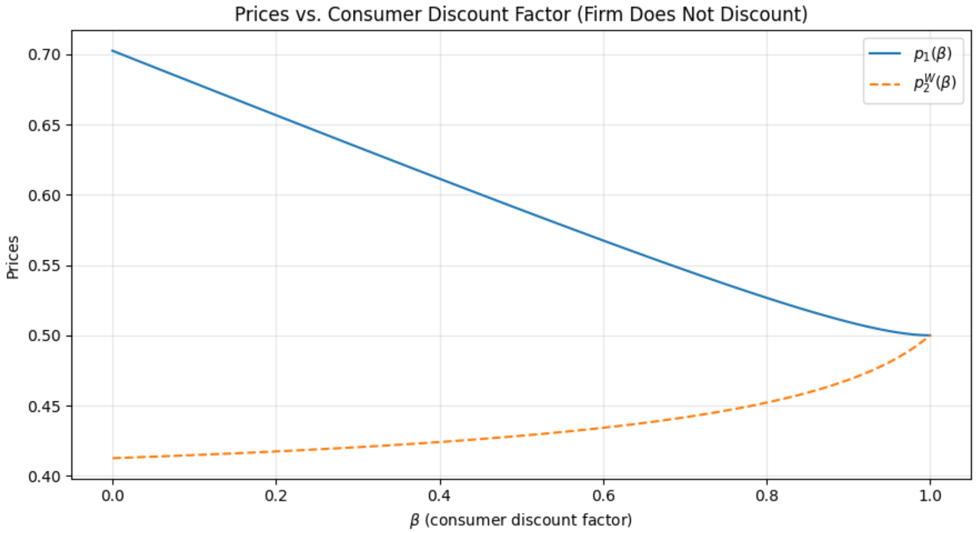

Figure 2: Two period game with two consumers. Consumer does not discount.

- Prove sublinear regret (or lack thereof) for stylized problem

References

8. APPENDIX

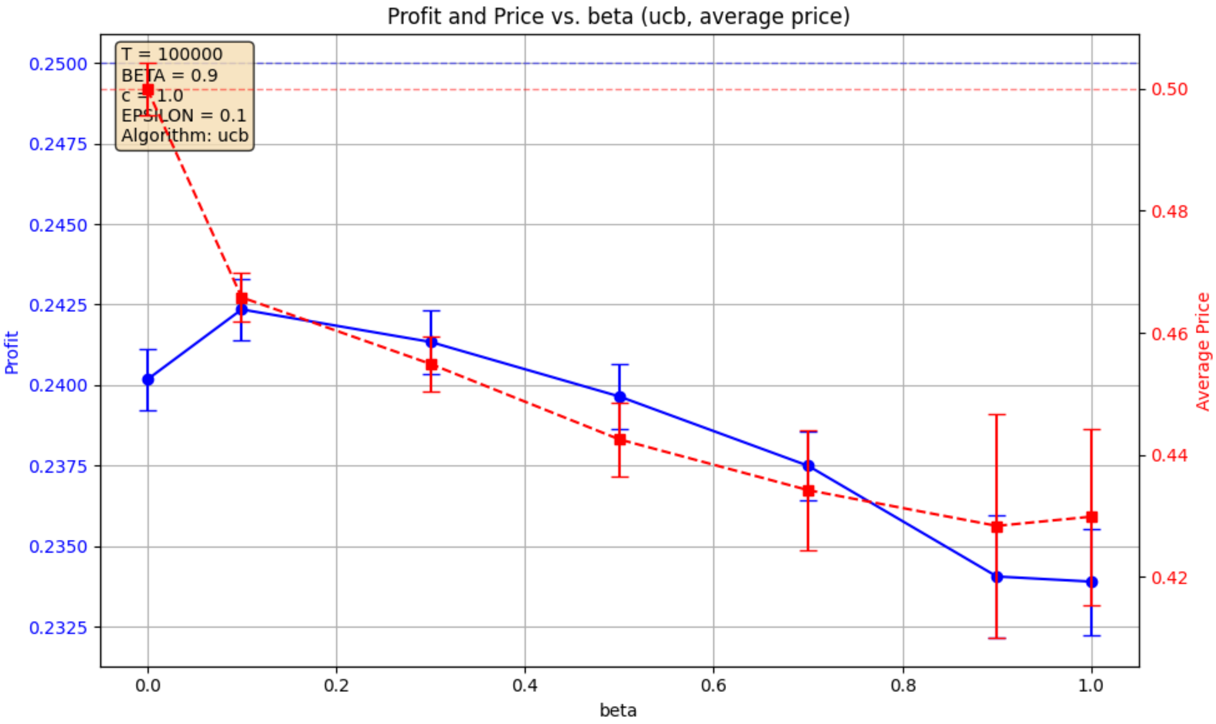

Figure 3: Simulation UCB1

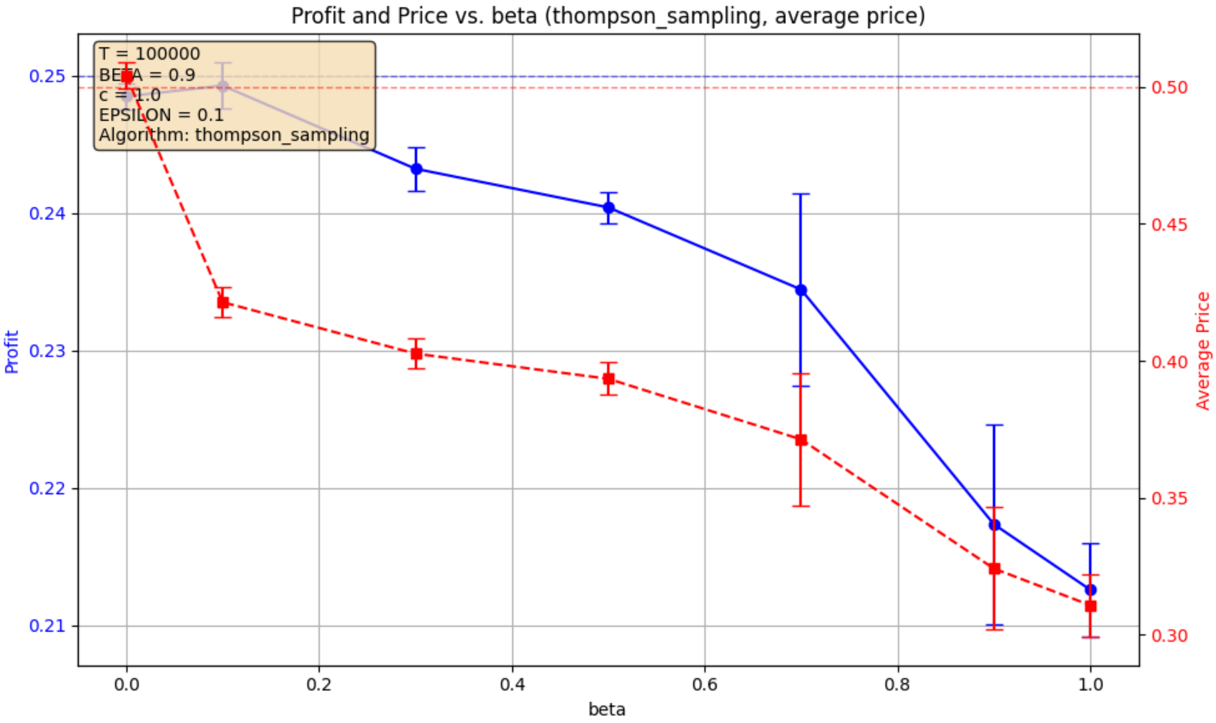

Figure 4: Simulation Thompson

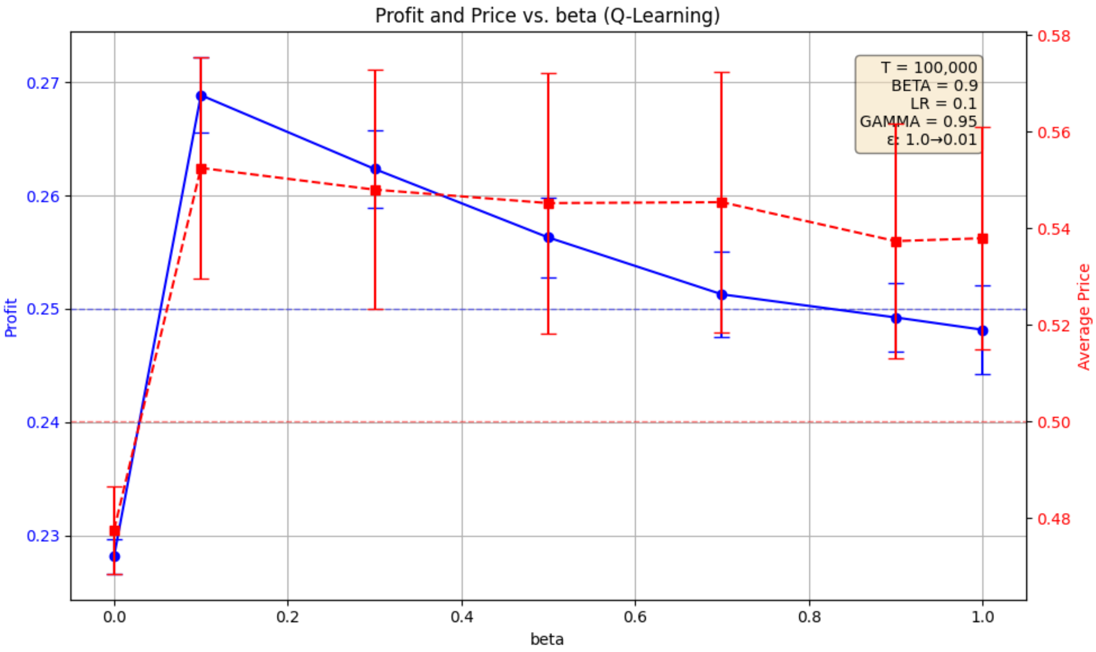

Figure 5: Simulation Q-learning