Learning to Price Against a Moving Target

Renato Paes Leme [1] **Balasubramanian Sivan** [1] **Yifeng Teng** [2] **Pratik Worah** [1]

Abstract

In the Learning to Price setting, a seller posts prices over time with the goal of maximizing revenue while learning the buyer's valuation. This problem is very well understood when values are stationary (fixed or iid). Here we study the problem where the buyer's value is a moving target, i.e., they change over time either by a stochastic process or adversarially with bounded variation. In either case, we provide matching upper and lower bounds on the optimal revenue loss. Since the target is moving, any information learned soon becomes out-dated, which forces the algorithms to keep switching between exploring and exploiting phases.

1. Introduction

Inspired by applications in electronic commerce, we study a problem where a seller repeatedly interacts with a buyer by setting prices for an item and observing whether the buyer purchases or not. These problems are characterized by two salient features: (i) binary feedback: we only observe if the buyer purchased or not, at the price we posted; (ii) discontinuous loss function: pricing just below the buyer's valuation incurs a small loss while pricing just above it incurs a large loss since it results in a no-sale.

This problem has been studied with many different assumptions on how the buyer valuation v_t changes over time: fixed over time and i.i.d. draws each round were studied in (Kleinberg & Leighton, 2003; Devanur et al., 2019; Cesa-Bianchi et al., 2019), deterministic contextual (Amin et al., 2014; Cohen et al., 2016; Lobel et al., 2017; Leme & Schneider, 2018; Liu et al., 2021), contextual with parametric noise (Javanmard & Nazerzadeh, 2019) and contextual with non-parametric noise (Shah et al., 2019; Krishnamurthy et al., 2020). All those models are stationary in the sense that the buyer's model is i.i.d. across time. The exceptions to this are algorithms that consider valuations that are drawn adversarially (Kleinberg & Leighton, 2003), but that work still compares with the best single price in hindsight. I.e., even though the buyer model is non-stationary, the benchmark still is.

Our main goal in this paper is to explore settings where both the buyer model and the benchmark are non-stationary. We will compare our revenue with the first-best benchmark, namely, the sum of the buyer's value at every single step. We will however assume that the buyer's valuation moves slowly.

Motivation Our main motivation for this study is online advertising. Display ads are mostly sold through first price auctions with reserve prices (Paes Leme et al., 2020). In many market segments, the auctions are thin, i.e., there is just one buyer, who bids just above the reserve when his value exceeds the reserve (to both pay as little as possible, and not reveal their true value) and doesn't bid otherwise. This scenario effectively offers just binary feedback, and also makes reserve price the only pricing tool (i.e., not much auction competition). To see why buyer value changes are typically slow, and unknown to the seller: the effective value of a buyer, even for two identical queries, is similar but not exactly the same due to factors such as remaining budget. A common scenario is that a buyer has a spend target stating a target θ_t of their daily budget to be spent by time t. Bids often become a function of the ratio between the actual spend and the target spend. The auction platform doesn't know the targets/bidding formula, but it can use the fact that both target and actual spend, and hence the bids, will change smoothly over time.

Another important motivation is to effectively price buyers who are learning about their own valuation. This is a common setup in finance (Shreve, 2004) where traders constantly acquire new information about the products they are trading, and update their valuations accordingly.

Our results and techniques are presented in Section 3 after we formally define our model in Section 2.

[*]Equal contribution [1]Google Research, New York, NY, USA [2]Department of Computer Sciences, University of Wisconsin-Madison, Madison, WI, USA. Correspondence to: Renato Paes Leme <renatoppl@google.com>, Balasubramanian Sivan <balusivan@google.com>, Yifeng Teng <yifengt@cs.wisc.edu>, Pratik Worah <pworah@google.com>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

Related Work Our work is situated in the intersection of two lines of work in online learning: online learning for pricing (discussed earlier in the introduction) and online learning with stronger benchmarks, such as tracking regret (Herbster & Warmuth, 2001; Luo & Schapire, 2015), adaptive regret (Hazan & Seshadhri, 2007), strongly adaptive online learning (Daniely et al., 2015) and shifting bandits (Foster et al., 2016; Lykouris et al., 2018). The difficulty in applying this line of work to pricing problems is that even when the valuation v_t changes slightly, the loss function itself will change dramatically for certain prices. Instead here, we exploit the special structure to the revenue loss to obtain better regret bounds.

There is another line of work that studies revenue maximization in the presence of evolving buyer values (Kakade et al., 2013; Pavan et al., 2014; Chawla et al., 2016). While all these works consider the cumulative value over time as benchmark, there are important differences, the first two papers have full feedback: they design mechanisms that solicit buyer bids. Chawla et al. shoot for simple pricing schemes yielding constant factor approximations, while we seek to obtain much closer to the optimal. Moreover, in their model the values evolve only when the buyer purchases the good.

2. Setting

General setting. A seller repeatedly interacts with a buyer over T time steps, offering to sell an item at each step. The buyer's value is $v_t \in [0,1]$ at time step $t \in [T]$, and is changing across time. The seller has no knowledge of the buyer's value at any time, not even at $t = 1$. At each time step t, the seller posts a price $p_t \in [0,1]$ to the buyer, and the buyer purchases the item if and only if she can afford to buy it, i.e. $p_t \leq v_t$. The binary signal

$$\sigma_t = \mathbf{1}\{v_t \geq p_t\} \in \{0,1\}$$

of whether the item is sold or not is the only feedback the seller obtains at each time step. The seller's objective is to maximize his the total revenue, i.e. $\textsc{Rev} = \sum_{t=1}^{T} p_t\sigma_t$.

Loss metrics. The benchmark we are comparing to is the *hindsight optimal* revenue we can get from the buyer, which is the sum of her value across all time periods: $\textsc{Opt} = \sum_{t=1}^{T} v_t$. The *revenue loss* at any time step t is defined by $\ell_{\mathrm{R}}(p_t, v_t) = v_t - p_t\sigma_t$, and the revenue loss of any price vector $\mathbf{p} = (p_1, \cdots, p_T)$ for value profile $\mathbf{v} = (v_1, \cdots, v_T)$ is defined by

$$\ell_{\mathrm{R}}(\mathbf{p}, \mathbf{v}) = \frac{1}{T}(\textsc{Opt} - \textsc{Rev}) = \frac{1}{T}\sum_{t=1}^{T}(v_t - p_t\sigma_t).$$

The *symmetric loss* at any time step t is defined by $\ell_1(p_t, v_t) = |v_t - p_t|$, and the total symmetric loss of any price vector $\mathbf{p} = (p_1, \cdots, p_T)$ for value profile $\mathbf{v} = (v_1, \cdots, v_T)$ is defined by

$$\ell_1(\mathbf{p}, \mathbf{v}) = \frac{1}{T}\sum_{t=1}^{T}|v_t - p_t|.$$

Intuitively, the revenue loss determines how much revenue we lose compared to a hindsight optimal selling strategy. The symmetric loss determines how close our price guesses are from the correct value vector of the buyer.

3. Our Results and Techniques

Next we describe our main results in this model. Our results will be given in terms of the changing rate, which we define as follows: consider a sequence of buyer valuations $v_1, v_2, \ldots, v_T$ and a sequence $\epsilon_1, \ldots, \epsilon_{T-1}$ (all $\epsilon_t \leq 1$). We say that a sequence of buyer valuations $\{v_t\}_{t=1..T}$ has changing rate $\{\epsilon_t\}_{t=1..T-1}$ whenever

$$|v_{t+1} - v_t| \leq \epsilon_t. \tag{1}$$

The average changing rate is $\bar{\epsilon} = \frac{1}{T-1}\sum_{t=1}^{T-1}\epsilon_t$. Our guarantees are a function of the average changing rate $\bar{\epsilon}$ but our algorithms are *agnostic* to both the changing rate sequence and its average (except in warmup section 4).

We will consider the problem in two environments:

1. Adversarial: an adaptive adversary chooses v_t's.
2. Stochastic: an adaptive adversary chooses a mean-zero distribution supported in $[-\epsilon_t, \epsilon_t]$ from which $v_{t+1} - v_t$ is drawn so that v_t is a bounded martingale. More generally, our results hold for any stochastic process satisfying Azuma's inequality.

We analyze three settings in total: symmetric loss in the adversarial environment, and revenue loss in both the adversarial and stochastic environments. In each of the three settings, we design our algorithms gradually starting from the simple case where the changing rate is fixed and known (warmup, Section 4), then fixed and unknown (Section 5)), and finally dynamic and unknown (Section 6)).

Dynamic, non-increasing changing rates In Section 6, where we study dynamic and unknown ϵ_t, we focus primarily on the case where the changing rates ϵ_t are non-increasing over time. This is motivated by situations where buyers use a learning algorithm with non-increasing learning rates (quite common) to determine their value, or using a controller to bid that stabilizes once the value approaches the ground truth. For symmetric loss alone (Theorem 6.4), we give guarantees for any sequence ϵ_t (i.e., not just non-increasing) but we get an additional $\log(T)$-factor in loss.

Our results For symmetric loss, we develop an algorithm with average loss $\tilde{O}(\bar{\epsilon})$ (Theorem 6.1) [and a slightly larger loss of $\tilde{O}(\bar{\epsilon}\log T)$ when the ϵ_t's are not necessarily non-increasing (Theorem 6.4)]. For revenue loss we show average loss of $\tilde{O}(\bar{\epsilon}^{1/2})$ in the adversarial setting (Theorem 6.2) and $\tilde{O}(\bar{\epsilon}^{2/3})$ in the stochastic setting (Theorem 6.3). Throughout, we use $\tilde{O}(f)$ to denote $O(f)\text{polylog}(1/\bar{\epsilon}) + o(1)$ where the $o(1)$ is with respect to T. Surprisingly, our loss bounds for none of the three settings in the general case of dynamic and unknown changing rates (Section 6) can be improved even if the changing rate is fixed and known (Section 4, Theorems 4.1, 4.2, 4.3). I.e., in our fairly general model, knowing the rate of change of valuation, or having the rate of change be fixed, don't provide much help in learning the valuation function to profitably price against it!

Our techniques *Step 1: fixed and known ϵ (Section 4):* Here, the algorithm keeps in each period a confidence interval $[\ell_t, r_t]$ containing v_t. If $r_t - \ell_t$ is small, we price at the lower endpoint ℓ_t, resulting in a larger interval $[\ell_{t+1}, r_{t+1}] = [\ell_t - \epsilon, r_t + \epsilon]$ in the next iteration. Once the interval is large enough, we decrease its size by a binary search to decrease and re-start the process. The algorithm thus keeps alternating between exploring and exploiting, where the length of the interval decides when we do what.

Step 2: fixed and unknown ϵ (Section 5): Here, we start guessing a very small value of ϵ, say $\hat{\epsilon} = 1/T$ and we behave as if we knew ϵ. If we detect any inconsistency with our assumption, we double our guess: $\hat{\epsilon} \leftarrow 2\hat{\epsilon}$. It is easy to detect when the value is below our confidence interval (since we always price at the lower point) but not so easy to detect when it is above. To address this issue, we randomly select certain rounds (with probability proportional to a power of our estimate $\hat{\epsilon}$) to be *checking rounds*. In those rounds, we price at the higher end of the interval.

Step 3: dynamic, non-increasing and unknown ϵ_t (Section 6): We again keep an estimate $\hat{\epsilon}$ like in Step 2, but now we adjust it in both directions. We increase $\hat{\epsilon}$ as in the previous step if we observe anything inconsistent with the guess. For the other direction, we *optimistically* halve the guess ($\hat{\epsilon} \leftarrow \hat{\epsilon}/2$) after we spend $1/\hat{\epsilon}$ periods with the same guess.

Comment on average versus total loss. Our results are all framed in terms of the average loss instead of the total loss, thus suppressing the factor T. To see why, note that even if we knew v_{t-1} exactly for each t, and even if values evolved stochastically, we would already incur an $O(\epsilon)$ loss per step leading to a total regret of $O(T\epsilon)$. Consequently, the main question is not the dependence on T (which is necessarily linear and cannot become any worse), but how much worse does the dependence on ϵ get, per time step. To see this in action, it is instructive to compare our algorithms with the algorithm in (Kleinberg & Leighton, 2003) which has sublinear loss with respect to a *fixed price benchmark*. For a fixed ϵ consider the periodic sequence v_t that starts at zero and increases by ϵ in each period reaching 1 and then decreases by ϵ in each period reaching 0 and starts climbing again. The first-best benchmark is $\sum_t v_t = \frac{T}{2} + O(\epsilon)$ while the best fixed price benchmark is $\frac{T}{4} + O(\epsilon)$. Our algorithm guarantees total revenue $T(\frac{1}{2} - \tilde{O}(\sqrt{\epsilon}))$, while Kleinberg & Leighton guarantee a revenue of $T\frac{1}{4} - O(\sqrt{T})$. In the supplementary material we show that for this example their algorithm indeed suffers a total loss of $\Omega(T)$ with respect to the first-best benchmark, while our algorithms suffer $T\tilde{O}(\sqrt{\epsilon})$, i.e., much better dependence on ϵ.

4. Warmup: Buyer's Changing Rate is Fixed, Known

We begin by studying the symmetric loss in the adversarial environment. The result is straightforward via a binary search algorithm that keeps track of a ***confidence interval*** $[\ell_t, r_t]$ that contains the true value v_t in each time step.

Theorem 4.1. *If the buyer has adversarial values and a fixed changing rate ϵ that is known to the seller, Algorithm 1 achieves symmetric loss $O(\epsilon)$. Further, no online algorithm can obtain symmetric loss less than $\Omega(\epsilon)$.*

Algorithm 1 Symmetric-loss minimizing algorithm for adversarial buyer with known changing rate ϵ

```
ℓ_1 ← 0
r_1 ← 1
for each time step t do
  Price at p_t ← (r_t + ℓ_t)/2
  if the current value v_t < p_t then
    ℓ_{t+1} ← max(0, ℓ_t − ε)
    r_{t+1} ← min(1, p_t + ε)
  else
    ℓ_{t+1} ← max(0, p_t − ε)
    r_{t+1} ← min(1, r_t + ε)
  end if
end for
```

Next, we extend the algorithm for symmetric loss to an optimal algorithm for revenue loss.

Theorem 4.2. *If the buyer has adversarial values and a fixed changing rate ϵ that is known to the seller, there exists an online pricing algorithm with revenue loss $\tilde{O}(\epsilon^{1/2})$. Further, no online algorithm can obtain a revenue loss less than $\Omega(\epsilon^{1/2})$.*

Proof. We assume that $\epsilon \geq \frac{1}{T}$. Otherwise, we can run the algorithm with $\epsilon' = \frac{1}{T}$ and get revenue loss $\tilde{O}((\epsilon')^{1/2}) = o(1) = \tilde{O}(\epsilon^{1/2})$.

The idea of the algorithm is as follows. The sale process starts with the seller not knowing the initial value v_1 of the buyer. However, since the buyer's value only changes by at most ϵ in each step, the seller can quickly locate the buyer's value within $O(\epsilon)$ error by binary search. Such a small confidence interval that contains the buyer's current value extends by ϵ in both upper bound and lower bound after each step. We propose an algorithm that repeatedly prices at the bottom of the confidence interval and re-locates the buyer's current value whenever the confidence interval becomes too wide.

Firstly, we have the following building-block algorithm that quickly returns the range of the buyer's value with additive accuracy $O(\epsilon)$. In the algorithm, timestamp t increases by one after any pricing query.

Algorithm 2 Locating the value of a buyer with changing rate ϵ to an interval of length $a \le 6\epsilon$

Input: current confidence interval $[\ell_t, r_t]$
Run Algorithm 1 until confidence interval $[\ell_t, r_t]$ satisfies $r_t - \ell_t < 4\epsilon$
Return $[\ell_t - \epsilon, r_t + \epsilon]$

The algorithm repeatedly does binary search until the confidence interval has length $O(\epsilon)$. If at the beginning we have a confidence interval of length b and finally we have a confidence interval of length a, thus the total number of steps needed is $O(\log \frac{b}{a})$, and the total loss of the algorithm is $O(\log \frac{b}{a})$ since the loss of each step is at most 1.

Next, we present the entire pricing algorithm for an adversarial buyer with changing rate ϵ using Algorithm 2 as a building block.

Algorithm 3 Revenue loss minimizing algorithm for adversarial buyer with known changing rate ϵ

for each phase $[t_0 + 1, t_0 + m]$ of length $m = \epsilon^{-1/2}$ **do**
  Apply Algorithm 2 to locate the current value of the buyer in an interval $[\ell_t, r_t]$ with length $\sqrt{\epsilon}$
  for each step t in the phase **do**
    Price at $p_t \leftarrow \ell_t$
    $[\ell_{t+1}, r_{t+1}] \leftarrow [\ell_t - \epsilon, r_t + \epsilon]$
  end for
end for

Now we analyze the revenue loss of the algorithm. In each phase of the algorithm with m time steps, we first initialize and locate the value of the buyer to a small range $\sqrt{\epsilon}$. The revenue loss incurred by Algorithm 2 is $O(\log \frac{1}{\epsilon}) = \tilde{O}(1)$ in each phase (actually $O(1)$ after the first phase since the length of confidence interval only needs to shrink by constant fraction). Then we price at the bottom of the confidence interval for $O(\sqrt{\epsilon})$ steps for $m = \sqrt{\epsilon}$ steps. Since the confidence interval $[\ell_t, r_t]$ expands by 2ϵ each time, it is equivalent to say we wait until the confidence interval has length $3\sqrt{\epsilon}$. The revenue loss from each step is $\le 3\sqrt{\epsilon}$ since the item is bought by the buyer every time, thus $O(\sqrt{\epsilon}) \cdot 3\sqrt{\epsilon} = O(\epsilon)$ in the entire phase; then we use binary search to narrow the confidence interval by $\frac{1}{2}$, which has revenue loss $O(1)$ since it takes only $O(1)$ steps in Algorithm 2. Each phase takes $\Theta(\sqrt{\epsilon})$ steps with loss $O(\epsilon)$, therefore there are $O(T\epsilon^{-1/2})$ phases in total with loss $O(T\epsilon^{-1/2}) \cdot O(\epsilon) = O(T\epsilon^{1/2})$. Thus the total loss of the algorithm is $\tilde{O}(T\epsilon^{1/2})$, with average revenue loss $\tilde{O}(\epsilon^{1/2})$.

We show that such loss is tight, that no online algorithm can obtain a revenue loss $o(\epsilon^{1/2})$. By Yao's minimax principle, to reason that there exists some adversarial buyer such that no randomized online algorithm can get revenue loss $o(\epsilon^{1/2})$, it suffices to show that there exists a random adversarial buyer such that no online deterministic algorithm can get such low revenue loss. The buyer's value sequence is predetermined as follows. In the beginning, the buyer has value $v_0 = \frac{1}{2}$. The entire time horizon $[T]$ is partitioned to $T\sqrt{\epsilon}$ phases, each with length $\epsilon^{-1/2}$. In each phase starting with time $t_0 + 1$ and ending with time $t_0 + \epsilon^{-1/2}$, the values of the buyer form a monotone sequence: with probability $\frac{1}{2}$, $v_{t_0+j} = v_{t_0} + j\epsilon, \forall j \in [\epsilon^{-1/2}]$; with probability $\frac{1}{2}$, $v_{t_0+j} = v_{t_0} - j\epsilon, \forall j \in [\epsilon^{-1/2}]$. For any deterministic algorithm with price p_t at each time, when any phase begins, the algorithm needs to decide the pricing strategy without knowing which value instance of the buyer will be realized. Let $\hat{t} \in [t_0 + 1, t_0 + \epsilon^{-1/2}]$ be the first time step in the phase such that $v_{t_0} - (t - t_0)\epsilon < p_t$.

If such $\hat{t}$ exists, then the revenue loss at time $\hat{t}$ is $v_{t_0} - (\hat{t} - t_0)\epsilon = v_{t_0} - O(\epsilon^{1/2})$ when the values of the buyer decrease in the phase, and this case happens with probability $\frac{1}{2}$. Then in expectation the revenue loss in this phase is $\Omega(v_{t_0}) - O(\epsilon^{1/2})$. Notice that the starting value v_{t_0} of each phase form a unbiased random walk sequence with step size $\epsilon^{1/2}$, since the buyer starts with $v_0 = \frac{1}{2}$, therefore with constant probability $v_{t_0} \ge \frac{1}{2}$. Thus we can also claim that the expected revenue loss in the phase is $\Omega(1)$. If such $\hat{t}$ does not exist, it means that the algorithm has identical information from both instances of the buyer in the entire phase, since the buyer can always afford to purchase the item. As $p_t \le v_{t_0} - (t - t_0)\epsilon$ for each time t, the revenue loss of the algorithm when the values of the buyer increase in the phase is at least $\sum_{t_0 < t \le t + \epsilon^{-1/2}} (v_{t_0} + (t - t_0)\epsilon - p_t) \ge \sum_{t_0 < t \le t + \epsilon^{-1/2}} 2(t - t_0)\epsilon = \Omega(1)$.

Therefore in both cases, the revenue loss in the phase is $\Omega(1)$ in this phase for any deterministic algorithm. As there are $T\epsilon^{1/2}$ phases, the total revenue loss from all phases is at least $\Omega(T\epsilon^{1/2})$. Thus no randomized algorithm can get average revenue loss $o(\epsilon^{1/2})$.

$\square$

If the buyer's value evolves stochastically across time, the stochasticity helps us incur less revenue loss. To be more specific, when the buyer's value is stochastic and forms a martingale, after every $\epsilon^{-2/3}$ steps, although the buyer's value can change by as large as $\epsilon^{-2/3} \cdot \epsilon = \epsilon^{1/3}$, with high probability the buyer's value only changes by $\epsilon^{2/3}$. Thus compared to Algorithm 3, we can extend each phase's length while maintaining a shorter confidence interval. We state the algorithm and the revenue loss below.

Algorithm 4 Revenue loss minimizing algorithm for stochastic buyer with known changing rate ϵ

$[\ell, r] \leftarrow [0, 1]$
for each phase $[t_0 + 1, t_0 + m]$ of length $m = \epsilon^{-2/3}$ **do**
  Apply Algorithm 2 to narrow down the confidence interval $[\ell, r]$ to length 6ϵ
  for each step t in the phase **do**
    Price at $p_t \leftarrow \ell - 4\epsilon^{2/3}\sqrt{\log \frac{1}{\epsilon}}$
  end for
end for

Theorem 4.3. *If the buyer has stochastic value and a fixed changing rate ϵ that is known to the seller, Algorithm 4 has revenue loss $\tilde{O}(\epsilon^{2/3})$. Furthermore, no online algorithm can obtain a revenue loss less than $\Omega(\epsilon^{2/3})$.*

5. Buyer's Changing Rate is Fixed, Unknown

In this section, we consider the case where the buyer's value changes by fixed rate $\epsilon_t = \epsilon$ that is unknown to the seller. As a warm-up, we first study the symmetric loss obtained by online prices. Such a problem lets us understand better how to deal with the unknown rate of value change, and the pricing algorithm can be extended to the case of revenue loss.

Theorem 5.1. *If the buyer has adversarial values and a fixed changing rate ϵ unknown to the seller, there exists an online pricing algorithm with a symmetric loss $\tilde{O}(\epsilon)$. Further, no online algorithm can obtain a symmetric loss less than $\Omega(\epsilon)$.*

Proof sketch. Compared to the case where ϵ is known to the seller, the seller can use a guess $\hat{\epsilon}$ to replace the true rate ϵ, and run the algorithms in the known ϵ setting. The algorithm starts with $\hat{\epsilon} = \frac{1}{T}$, and doubles $\hat{\epsilon}$ whenever the algorithm detects that the true changing rate exceeds $\hat{\epsilon}$.

The algorithm deals with three time steps $t, t+1, t+2$ at each time, and always maintain a confidence interval $[\ell_t, r_t]$ of v_t at the beginning of time step t, such that $\hat{\epsilon} > \epsilon$,

Algorithm 5 Symmetric-loss minimizing algorithm for adversarial buyer with unknown changing rate ϵ

$[\ell_1, r_1] \leftarrow [0, 1]$
$\hat{\epsilon} \leftarrow \frac{1}{T}$
while $\hat{\epsilon} < \frac{1}{2}$ **do**
  for each three consecutive time steps $t, t+1, t+2$ **do**
    Set price $p_t \leftarrow \ell_t$
    Set price $p_{t+1} \leftarrow r_t + \hat{\epsilon}$
    Set price $p_{t+2} \leftarrow \frac{\ell_t + r_t}{2}$
    if the seller finds $v_t < \ell$ or $v_{t+1} \geq r_t + \hat{\epsilon}$ **then**
      $\hat{\epsilon} \leftarrow 2\hat{\epsilon}$
      break
    end if
    if $v_{t+2} < p_{t+2}$ **then**
      $\ell_{t+3} \leftarrow \ell_t - 3\hat{\epsilon}$, $r_{t+3} \leftarrow p_{t+2} + \hat{\epsilon}$
    else
      $\ell_{t+3} \leftarrow p_{t+2} - \hat{\epsilon}$, $r_{t+3} \leftarrow r + 3\hat{\epsilon}$
    end if
  end for
end while

$v_t \in [\ell_t, r_t]$ always holds. Time steps t and $t+1$ are "checking steps", to check whether the current three steps v_t , v_{t+1} and v_{t+2} incur too much loss. In particular, if $\hat{\epsilon} \geq \epsilon$, then $v_t \geq p_t$ and $v_{t+1} < p_{t+1}$ will always hold, and the symmetric loss is $O(\epsilon)$ per step from Theorem 4.1. On the other hand, in an iteration with $\hat{\epsilon} < \epsilon$, if $v_t \geq p_t = \ell_t$ and $v_{t+1} < p_{t+1} = r_t + \hat{\epsilon}$ always hold, then $v_t, v_{t+1}, v_{t+2} \in [\ell_t - O(\epsilon), r_t + O(\epsilon)]$ since the value of the buyer only changes by at most ϵ per step. Thus in a round of three consecutive time steps $t, t+1, t+2$, if $v_t \geq p_t$ and $v_{t+1} < p_{t+1}$, the symmetric loss from this round of three time steps is $O(r_t - \ell_t + \epsilon)$. Notice that there are only $O(\log T)$ different $\hat{\epsilon} < \epsilon$. For each $\hat{\epsilon}$ the initialization has $O(1)$ symmetric loss, while the stable length of confidence interval $r_t - \ell_t = O(\hat{\epsilon}) = O(\epsilon)$, thus the average symmetric loss of each step is $\tilde{O}(\epsilon)$.

$\square$

Using a similar technique of checking steps as in the case of symmetric loss, we can obtain the same revenue loss as in the setting where the seller knows the changing rate, no matter whether the buyer has adversarial or stochastic value.

Theorem 5.2. *If the buyer has adversarial values and a fixed changing rate ϵ unknown to the seller, there exists an online pricing algorithm with revenue loss $\tilde{O}(\epsilon^{1/2})$. Further, no online algorithm can obtain a revenue loss less than $\Omega(\epsilon^{1/2})$.*

Proof. The tightness $\Omega(\epsilon^{1/2})$ result is shown in Theorem 4.2, so we only need to construct an algorithm with revenue loss $\tilde{O}(\epsilon^{1/2})$. We want to apply the same technique of "checking steps" as in the pricing algorithm for symmetric loss. Recall that the checking steps in Algorithm 5 repeatedly price at the upper bound and the lower bound of the current confidence interval of the value of the buyer to make sure the buyer's value is not too far away from the confidence bound. However, when minimizing revenue loss, the seller cannot afford to frequently check the upper bound of the confidence interval, since each time the buyer is very likely to reject the item and incurs a huge revenue loss. The solution to such a problem is that we can add one checking step to each phase of Algorithm 3. By pricing at the lower bound ℓ_t of the confidence interval at time t and the upper bound r_{t+1} of the confidence interval at time $t+1$, the seller can know whether v_{t-1} is far away from our confidence interval $[\ell_{t-1}, r_{t-1}]$ for it. Thus for a phase with m time steps and k "bad steps" when the buyer's value is far away from the confidence interval, a random checking step can detect a bad step with probability $\frac{k}{m}$. This means that after $O(m)$ bad steps in expectation, a bad step will be detected and the algorithm will move to the next iteration with $\hat{\epsilon}$ doubled. The algorithm is stated as follows.

Algorithm 6 Revenue loss minimizing algorithm for adversarial buyer with unknown changing rate ϵ

```
ε̂ ← 1/T
while ε̂ < 1/2 do
  for each phase [t_0 + 1, t_0 + m_ε̂] of length m_ε̂ = ε̂^{-1/2} do
    Apply Algorithm 2 to locate the current value of the buyer in an interval [ℓ_t, r_t] with length √ε̂
    Randomly select a t* ∈ [t_0 + 1, t_0 + m_ε̂]
    for each step t in the phase do
      if t ≠ t* then
        Price at p_t ← ℓ_t
        if v_t < p_t then
          ε̂ ← 2ε̂ and break (terminate the phase)
        end if
      else
        Price at p_t ← r_t
        if v_t ≥ p_t then
          ε̂ ← 2ε̂ and break (terminate the phase)
        end if
      end if
      [ℓ_{t+1}, r_{t+1}] ← [ℓ_t − ε, r_t + ε]
    end for
  end for
end while
```

The algorithm first guesses $\hat{\epsilon} = \frac{1}{T}$, and doubles $\hat{\epsilon}$ whenever the algorithm detects a piece of evidence of $\hat{\epsilon}$ being smaller than the true ϵ. In particular, the algorithm maintains confidence bound $[\ell_t, r_t]$ that may contain the current value v_t of the buyer. When $\hat{\epsilon}$ first becomes larger than ϵ (thus at most 2ϵ), the algorithm will run smoothly without triggering any break statement since $v_t \in [\ell_t, r_t]$ always holds. The revenue loss is $\tilde{O}(\sqrt{\hat{\epsilon}}) = \tilde{O}(\sqrt{\epsilon})$ as analyzed in Theorem 4.2.

In an iteration when $\hat{\epsilon} < \epsilon$, firstly there is an additional $O(\log \frac{1}{\hat{\epsilon}}) = \tilde{O}(1)$ loss for binary search initialization of the confidence interval compared to Algorithm 3 for the fixed ϵ setting. Let bad event $\mathcal{E}_t$ denote "$v_t \geq r_t + 2\epsilon$ or $v_t < \ell_t$". If bad events never happen, the additional loss is at most 2ϵ per step (thus $T\epsilon$ in total), since in the analysis of Algorithm 3 it has confidence interval $[\ell_t, r_t]$ rather than $[\ell_t, r_t + 2\epsilon]$ here. In a phase with time steps $[t_0+1, t_0+m_{\hat{\epsilon}}]$, if an event $\mathcal{E}_t$ happens, then there are two cases. If $v_t < \ell_t$, then it is detected when $p_t = \ell_t$, which almost surely happens. If $v_t > r_t + 2\epsilon$, then it is detected when $p_{t+1} = r_{t+1}$ i.e. $t^* = t$, since $v_{t+1} \geq v_t - \epsilon > r_t + \epsilon = r_{t+1}$. Therefore, since t^* is randomly selected from $[t_0+1, t_0+m_{\hat{\epsilon}}]$, if k events in $\mathcal{E}_{t_0+1}, \cdots, \mathcal{E}_{t_0+m_{\hat{\epsilon}}}$ happens, with probability $\frac{k}{m_{\hat{\epsilon}}}$ a bad event gets detected. Thus, in an iteration with fixed $\hat{\epsilon} < \epsilon$, when a bad event is detected, in expectation m bad events have occurred. Each bad event will result in additional revenue loss at most 1, thus $\tilde{O}(m_{\hat{\epsilon}}) = O(\hat{\epsilon}^{-1/2})$. The total contribution of revenue loss from bad events is at most $\sum_{i=0}^{\log T} \left(\frac{2^i}{T}\right)^{-1/2} = O(T^{1/2}) = To(1)$. To summarize, the total revenue loss of the algorithm is $\tilde{O}(T\epsilon^{1/2})$ for Algorithm 3 with known ϵ, plus the binary search cost $\tilde{O}(1)$ for locating the position of v_t in each iteration of different $\hat{\epsilon}$, plus the additional cost $O(T\epsilon)$ for having a slightly larger confidence interval in good events than Algorithm 3, plus a total revenue loss of $T^{1/2}$ from the bad events. Sum up all the costs above we show that the total revenue loss of Algorithm 6 is $T\tilde{O}(\epsilon^{1/2})$ for all time steps, thus $\tilde{O}(\epsilon^{1/2})$ on average.

□

When the buyer has stochastic value, we can modify Algorithm 6 for an adversarial buyer such that in each phase is replaced by a phase in Algorithm 4, with the normal pricing step $t \neq t^*$ pricing at $p_t = \ell_{t_0} - \tilde{O}(\hat{\epsilon}^{-1/3})$, and each checking step t^* at price $p_{t^*} = r_{t_0} + \tilde{O}(\hat{\epsilon}^{-1/3})$. The analysis is almost identical to Theorem 5.2.

Theorem 5.3. *If the buyer has stochastic value and a fixed changing rate ϵ unknown to the seller, there exists an online pricing algorithm with revenue loss $\tilde{O}(\epsilon^{2/3})$. Further, no online algorithm can obtain a revenue loss less than $\Omega(\epsilon^{2/3})$.*

6. Buyer's Changing Rate is Dynamic, Unknown

In this section, we study a more complicated setting where the buyer's value changes in a more dynamic way. In particular, $|v_{t+1} - v_t|$ are upper bounded by possibly different non-increasing ϵ_t that are unknown to the seller.

For the symmetric-loss minimization problem with an adversarial buyer, we show that when ϵ_t are non-increasing, the seller can still achieve a symmetric loss of $\tilde{O}(\bar{\epsilon})$ as in the case of fixed ϵ. The algorithm and its analysis have the same structure as the revenue loss minimization problem and is omitted here.

Theorem 6.1. *If the buyer has adversarial values and non-increasing changing rates ϵ_t unknown to the seller, then there exists an online pricing algorithm with symmetric loss $\tilde{O}(\bar{\epsilon})$. Furthermore, no online algorithm can obtain a symmetric loss less than $\Omega(\bar{\epsilon})$.*

For the revenue loss minimization problem for an adversarial buyer, we can also recover the results in previous sections, even when ϵ_t are unknown to the seller. We describe the result and the algorithm in detail here.

Theorem 6.2. *If the buyer has adversarial values and non-increasing changing rates ϵ_t unknown to the seller, there exists an online pricing algorithm with revenue loss $\tilde{O}(\bar{\epsilon}^{1/2})$, here $\bar{\epsilon} = \frac{1}{T}\sum_{t=1}^{T}\epsilon_t$. Further, no online algorithm can obtain a revenue loss less than $\Omega(\bar{\epsilon}^{1/2})$.*

Proof. The $\Omega(\bar{\epsilon}^{1/2})$ tightness result has been shown in Theorem 4.2 with all ϵ_t being identical. Now we show that there exists an algorithm with revenue loss $\tilde{O}(\bar{\epsilon}^{1/2})$.

When ϵ_t decreases, the seller needs to detect such a trend timely, otherwise the loss of each time step is going to be not comparable to $\sum_t \epsilon_t$. We propose the following algorithm for the seller, that repeatedly guesses the current level of changing rate at each time step. The algorithm starts with guessing $\hat{\epsilon} = \frac{1}{2}$ being an estimate of ϵ_t, and reduces the value of the guess $\hat{\epsilon}$ by a factor of $\frac{1}{2}$ if in several time steps the algorithm cannot find any evidence supporting $\hat{\epsilon} < \epsilon_t$. Whenever the algorithm finds evidence that supports $\hat{\epsilon} < \epsilon_t$, the algorithm repeatedly doubles $\hat{\epsilon}$ and updates the confidence interval according to the new $\hat{\epsilon}$, until the evidence of $\hat{\epsilon} < \epsilon_t$ disappears. Such a dynamic update of $\hat{\epsilon}$ keeps the revenue loss bounded.

To be more specific, $\hat{\epsilon}$ decreases by a factor of $\frac{1}{2}$ if the seller has not observed any evidence of $\hat{\epsilon} < \epsilon_t$ for *long enough time*. In particular, the algorithm tries to run $\hat{\epsilon}^{-1/2}$ identical phases in Algorithm 6, and will halve $\hat{\epsilon}$ when the buyer passes all checking steps. The algorithm is described in Algorithm 7.

Now we analyze the performance of the algorithm. Partition the time horizon $[T]$ to $\log T$ intervals $I_1, \cdots, I_{\log T}$, such that for each time interval I_i and time $t \in I_i$, $\epsilon_t \in (2^{-i}, 2^{-i+1}]$. Let $\epsilon_i^* = 2^{-i+1}$. We argue that in time interval I_i the total revenue loss is $\tilde{O}((\epsilon_i^*)^{-1/2} + |I_i|(\epsilon_i^*)^{1/2})$.

In interval I_i the algorithm may start with $\hat{\epsilon} > \epsilon_i^*$, and then gradually decreases to reach $\hat{\epsilon} = \epsilon_i^*$ and never become larger than ϵ_i^* later. In this process, the revenue loss is $\tilde{O}(1)$ in each phase as shown in the proof of Theorem 5.2, thus $\tilde{O}(\hat{\epsilon}^{-1/2})$ loss for every value $\hat{\epsilon} > \epsilon_i^*$ and at most $\sum_{\hat{\epsilon}>\epsilon_i^*} \tilde{O}(\hat{\epsilon}^{-1/2}) = \tilde{O}((\epsilon_i^*)^{-1/2})$ in total.

Now we show that after $\hat{\epsilon}$ reaches ϵ_i^*, the revenue loss is $\tilde{O}((\epsilon_i^*)^{1/2})$ per step on average. First we study the revenue loss from each $\hat{\epsilon}^{-1/2}$ phases with changing rate $\hat{\epsilon}$. The same as in previous proofs, a piece of "bad evidence", or a piece of evidence of $\hat{\epsilon} < \epsilon$ is the event of $v_t < p_t$ in a non-checking step $t \neq t^*$ or $v_t \geq p_t$ in a checking step $t = t^*$. If no evidence of $\hat{\epsilon} < \epsilon$ is detected, then the revenue loss is at most some constant $c = \tilde{O}(1)$ in each phase, thus $c\hat{\epsilon}^{-1/2}$ for the $\hat{\epsilon}^{-1/2}$ phases with $\hat{\epsilon}^{-1}$ time steps. We also observe that if we run the algorithm with changing rate ϵ_i^*, the revenue loss of such $\hat{\epsilon}^{-1}$ time steps is going to be $c(\epsilon_i^*)^{1/2}$ per step thus $c\hat{\epsilon}^{-1}(\epsilon_i^*)^{1/2}$ in total.

We argue that the per-step revenue loss in I_i is at most $2c(\epsilon_i^*)^{1/2}$. In $\hat{\epsilon}^{-1/2}$ phases where no bad evidence is found, the algorithm actually gets $c(\hat{\epsilon}^{-1}(2\epsilon_i^*)^{1/2} - \hat{\epsilon}^{1/2}) > c\hat{\epsilon}^{-1}(\epsilon_i^*)^{1/2} > c\hat{\epsilon}^{-1/2}$ less revenue loss than the expected benchmark $(2c(\epsilon_i^*)^{1/2}$ per step). In any phase with estimated changing rate $\hat{\epsilon}$ where a piece of bad evidence is

Algorithm 7 Revenue loss minimizing algorithm for adversarial buyer with unknown decreasing changing rate ϵ_t

```
ε̂ ← 1/2
while true do
  for ε̂^{-1/2} phases of length m_ε̂ = ε̂^{-1/2} do
    At the beginning of phase [t_0 + 1, t_0 + m_ε̂], apply
    Algorithm 2 to locate the current value of the buyer
    in an interval [ℓ_t, r_t] with length √ε̂
    Randomly select a t* ∈ [t_0 + 1, t_0 + m_ε̂]
    for each step t in the phase do
      if t ≠ t* then
        Price at p_t ← ℓ_t
        if v_t < p_t then
          ε̂ ← 2ε̂ and go back to the beginning of the
          while loop (terminate the ε̂^{-1/2} phases)
        end if
      else
        Price at p_t ← r_t
        if v_t ≥ p_t then
          ε̂ ← 2ε̂ and go back to the beginning of the
          while loop (terminate the ε̂^{-1/2} phases)
        end if
      end if
      [ℓ_{t+1}, r_{t+1}] ← [ℓ_t − ε, r_t + ε]
    end for
  end for
  ε̂ ← ε̂/2 if ε̂ > 1/T
end while
```

found, as shown in the analysis of Algorithm 6, in expectation $m_{\hat{\epsilon}} = \hat{\epsilon}^{-1/2}$ steps with value out of confidence bound has occurred, and contributes at most $\hat{\epsilon}^{-1/2}$ total additional revenue loss more than the normal $2c(\epsilon_i^*)^{1/2}$ loss per step. Therefore, every time the algorithm goes through $\hat{\epsilon}^{-1/2}$ phases without bad evidence, the algorithm has at least $c\hat{\epsilon}^{-1/2}$ less revenue loss than expected; every time the algorithm with estimated changing rate $\hat{\epsilon}$ finds a phase with a piece of bad evidence, the algorithm has at most $c\hat{\epsilon}^{-1/2}$ more revenue loss than expected. Observe that in each iteration with $\hat{\epsilon}$ decreases by $\frac{1}{2}$ no bad evidence is detected, and bad evidence is found in each iteration with $\hat{\epsilon}$ getting doubled. Thus the number of iterations with no bad evidence being detected is at least the number of iterations with bad evidence found, which means that the algorithm has no more revenue loss than the expected $2c(\epsilon_i^*)^{1/2}$ per step. To summarize, in I_i after $\hat{\epsilon}$ reaches ϵ_i^*, the revenue loss is $\tilde{O}((\epsilon_i^*)^{1/2})$ per step.

Above reasoning shows that in each time interval I_i, the total revenue loss is $\tilde{O}((\epsilon_i^*)^{-1/2} + |I_i|(\epsilon_i^*)^{1/2})$. Sum up over all i, the total revenue loss of all time steps is

$$\begin{aligned} & \sum_{i \le \log T} \tilde{O}((\epsilon_i^*)^{-1/2} + |I_i|(\epsilon_i^*)^{1/2}) \\ = \quad & \tilde{O}(T^{1/2}) + \tilde{O}(1) \sum_i |I_i|(\epsilon_i^*)^{1/2} \\ \le \quad & To(1) + \tilde{O}(1) \sum_i |I_i| \bar{\epsilon}^{1/2} = \tilde{O}(T\bar{\epsilon}^{1/2}), \end{aligned}$$

Here the inequality is by Cauchy-Schwarz. Thus the average revenue loss of each time step is $\tilde{O}(\bar{\epsilon}^{1/2})$.

□

Such a result can also be extended for a stochastic buyer.

Theorem 6.3. *If the buyer has stochastic value and non-increasing changing rate ϵ_t unknown to the seller, then there exists an online pricing algorithm with revenue loss $\tilde{O}(\bar{\epsilon}^{2/3})$, here $\bar{\epsilon} = \frac{1}{T}\sum_{t=1}^{T} \epsilon_t$. Furthermore, no online algorithm can obtain a revenue loss less than $\Omega(\bar{\epsilon}^{2/3})$.*

For the revenue loss minimization problem, it is hard to obtain positive results when the changing rates ϵ_t are arbitrary, since setting a price slightly higher than the true value in a step can result in a huge revenue loss. Surprisingly, even if ϵ_t for each time step can change arbitrarily, we can still achieve the $\tilde{O}(\bar{\epsilon})$ loss in previous sections, only losing a tiny $O(\log T)$ factor.

Theorem 6.4. *If the buyer has adversarial values and dynamic changing rate ϵ_t unknown to the seller, there exists an online pricing algorithm with symmetric loss $\tilde{O}(\bar{\epsilon}\log T)$ for $\bar{\epsilon} = \frac{1}{T}\sum_{t\in[T]} \epsilon_t$. Further, no online algorithm can obtain a symmetric loss less than $\Omega(\bar{\epsilon})$.*

Proof sketch. Suppose for a moment that the algorithm is allowed to set multiple pricing queries for a single time step. The algorithm maintains a correct confidence interval $[\ell_t, r_t]$ that contains the value v_t of the buyer at each time step. At time $t+1$, the seller does not know the exact value change $v_{t+1} - v_t$. Furthermore, she also does not know a bound of the value change $\epsilon_t \ge |v_{t+1} - v_t|$. However, the seller can try to price at $\ell_t - \delta_j$ and $r_t + \delta_j$ repeatedly for every j and $\delta_j = 2^j T^{-1}$. When j has increased such that the algorithm finds that $\ell_t - \delta_j < v_{t+1} < r_t + \delta_j$, the seller can then price at $\frac{\ell_t + r_t}{2}$ to get a new correct confidence interval $[\ell_{t+1}, r_{t+1}] \leftarrow [\ell_t - \delta_j, \frac{\ell_t + r_t}{2}]$ or $[\frac{\ell_t + r_t}{2}, r_t + \delta_j]$. Such an algorithm identifies the change of each step accurately and has $\tilde{O}(\bar{\epsilon})$ symmetric loss.

Algorithm 8 Symmetric-loss minimizing algorithm for adversarial buyer with unknown dynamic changing rate ϵ_t

```
[ℓ_1, r_1] ← [0, 1]
Let t_1 + 1 = 1 be the starting time of the first phase
for each phase i of time interval [t_i + 1, t_{i+1}] with to-be-determined stopping time t_{i+1} do
    Let t_i + 1 be the starting time step of the phase
    for each integer j ≥ 0 do
        ε̂ ← δ_j = 2^j T^{-1}
        Set price p_{t_i+2j+1} ← ℓ_{t_i+1} − δ_j
        Set price p_{t_i+2j+2} ← r_{t_i+1} + δ_j
        if the seller finds v_{t_i+2j+1} ≥ p_{t_i+2j+1} and v_{t_i+2j+2} < p_{t_i+2j+2} then
            break (from this for loop)
        end if
    end for
    t_{i+1} ← t_i + 2j + 3
    Set p_{t_{i+1}} ← (ℓ_{t_i+1} + r_{t_i+1}) / 2
    if v_{t_{i+1}} < p_{t_{i+1}} then
        [ℓ_{t_{i+1}+1}, r_{t_{i+1}+1}] ← [ℓ_{t_i+1} − δ_j, p_{t_{i+1}}]
    else
        [ℓ_{t_{i+1}+1}, r_{t_{i+1}+1}] ← [p_{t_{i+1}}, r_{t_i+1} + δ_j]
    end if
end for
```

However, we are not allowed to have multiple pricing queries for the same value. The key observation is that when we serialize the pricing queries in such an algorithm with at most k queries per step, the symmetric loss only increases by a factor of $O(k)$. Since the parallel algorithm has at most $O(\log T)$ pricing queries in each step, the serialized algorithm's symmetric loss only increases by $O(\log T)$. Another specific example of such serialization is Algorithm 5 for minimizing symmetric loss in the unknown fixed changing rate setting. Each three consecutive steps $t, t+1, t+2$ can be viewed as the serialization of a single step with three price queries $\ell_t - \epsilon$, $r_t + \epsilon$ and $\frac{\ell_t + r_t}{2}$. The analysis of the serialized Algorithm 8 for this unknown changing ϵ_t setting can be viewed as a generalization of the analysis of

Algorithm 5 for the unknown fixed changing rate setting.

□

Acknowledgements

We thank the reviewers for their valuable feedback. Yifeng Teng was supported in part by NSF grants CCF-1617505 and SHF-1704117. Part of this work was done when Yifeng Teng was an intern at Google.

References

Amin, K., Rostamizadeh, A., and Syed, U. Repeated contextual auctions with strategic buyers. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pp. 622–630, 2014.

Cesa-Bianchi, N., Cesari, T., and Perchet, V. Dynamic pricing with finitely many unknown valuations. In *Algorithmic Learning Theory*, pp. 247–273. PMLR, 2019.

Chawla, S., Devanur, N. R., Karlin, A. R., and Sivan, B. Simple pricing schemes for consumers with evolving values. In Krauthgamer, R. (ed.), *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pp. 1476–1490. SIAM, 2016.

Cohen, M. C., Lobel, I., and Paes Leme, R. Feature-based dynamic pricing. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, pp. 817–817. ACM, 2016.

Daniely, A., Gonen, A., and Shalev-Shwartz, S. Strongly adaptive online learning. In *International Conference on Machine Learning*, pp. 1405–1411. PMLR, 2015.

Devanur, N. R., Peres, Y., and Sivan, B. Perfect bayesian equilibria in repeated sales. *Games Econ. Behav.*, 118: 570–588, 2019.

Foster, D. J., Li, Z., Lykouris, T., Sridharan, K., and Tardos, É. Learning in games: robustness of fast convergence. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 4734–4742, 2016.

Hazan, E. and Seshadhri, C. Adaptive algorithms for online decision problems. In *Electronic colloquium on computational complexity (ECCC)*, volume 14, 2007.

Herbster, M. and Warmuth, M. K. Tracking the best linear predictor. *Journal of Machine Learning Research*, 1(281-309):10–1162, 2001.

Javanmard, A. and Nazerzadeh, H. Dynamic pricing in high-dimensions. *Journal of Machine Learning Research*, 20 (9):1–49, 2019. URL http://jmlr.org/papers/v20/17-357.html.

Kakade, S. M., Lobel, I., and Nazerzadeh, H. Optimal dynamic mechanism design and the virtual-pivot mechanism. *Oper. Res.*, 61(4):837–854, 2013.

Kleinberg, R. and Leighton, T. The value of knowing a demand curve: Bounds on regret for online posted-price auctions. In *Foundations of Computer Science, 2003. Proceedings. 44th Annual IEEE Symposium on*, pp. 594–605. IEEE, 2003.

Krishnamurthy, A., Lykouris, T., Podimata, C., and Schapire, R. Contextual search in the presence of irrational agents. *arXiv preprint arXiv:2002.11650*, 2020.

Leme, R. P. and Schneider, J. Contextual search via intrinsic volumes. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pp. 268–282, 2018. doi: 10.1109/FOCS.2018.00034. URL https://doi.org/10.1109/FOCS.2018.00034.

Liu, A., Leme, R. P., and Schneider, J. Optimal contextual pricing and extensions. pp. 1059–1078, 2021. doi: 10.1137/1.9781611976465.66. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611976465.66.

Lobel, I., Leme, R. P., and Vladu, A. Multidimensional binary search for contextual decision-making. *Operations Research*, 2017.

Luo, H. and Schapire, R. E. Achieving all with no parameters: Adanormalhedge. In *Conference on Learning Theory*, pp. 1286–1304. PMLR, 2015.

Lykouris, T., Sridharan, K., and Tardos, É. Small-loss bounds for online learning with partial information. In *Conference on Learning Theory*, pp. 979–986. PMLR, 2018.

Paes Leme, R., Sivan, B., and Teng, Y. Why do competitive markets converge to first-price auctions? In *Proceedings of The Web Conference 2020*, pp. 596–605, 2020.

Pavan, A., Segal, I., and Toikka, J. Dynamic mechanism design: A myersonian approach. *Econometrica*, 82(2):601–653, 2014. doi: https://doi.org/10.3982/ECTA10269. URL https://onlinelibrary.wiley.com/doi/abs/10.3982/ECTA10269.

Shah, V., Johari, R., and Blanchet, J. Semi-parametric dynamic contextual pricing. 32:2363–2373, 2019. URL https://proceedings.

neurips.cc/paper/2019/file/363763e5c3dc3a68b399058c34aecf2c-Paper.pdf.

Shreve, S. E. *Stochastic calculus for finance II: Continuous-time models*, volume 11. Springer Science & Business Media, 2004.