Learning to Price Against a Moving Target

Renato Paes Leme
Google Research
renatoppl@google.com

Balasubramanian Sivan
Google Research
balusivan@google.com

Yifeng Teng
UW-Madison
yifengt@cs.wisc.edu

Pratik Worah
Google Research
pworah@google.com

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 [13, 7, 3], deterministic contextual [1, 5, 17, 15, 16], contextual with parametric noise [11] and contextual with non-parametric noise [22, 14]. 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 [13], 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 [20]. 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 [23] 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.

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 [10, 18], adaptive regret [9], strongly adaptive online learning [6] and shifting bandits [8, 19]. 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 [12, 21, 4]. 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. $\text{REV} = \sum_{t=1}^{T} p_t \sigma_t$.

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: $\text{OPT} = \sum_{t=1}^{T} v_t$. The *revenue*

2

loss at any time step t is defined by $\ell_R(p_t, v_t) = v_t - p_t\sigma_t$, 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

$$\ell_R(\mathbf{p}, \mathbf{v}) = \frac{1}{T}(\text{OPT} - \text{REV}) = \frac{1}{T}\sum_{t=1}^{T}(v_t - p_t\sigma_t).$$

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.

3

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 $O(f)\mathrm{polylog}(1/\bar{\epsilon}) + o(1)$ 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 [13] 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 $T(\frac{1}{2} - O(\sqrt{\epsilon}))$, while Kleinberg and 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, there is an online pricing algorithm that achieves symmetric loss $O(\epsilon)$. Further, no online algorithm can obtain symmetric loss less than $\Omega(\epsilon)$.*

Proof. Firstly, consider the value sequence to be an unbiased random walk with step size ϵ, and starting point $v_1 = \frac{1}{2}$. The symmetric loss from each time step t is $\Omega(\epsilon)$ even if v_{t-1} is known to the seller, since the seller does not know whether $v_t = v_{t-1} + \epsilon$ or $v_t = v_{t-1} - \epsilon$. Thus no online algorithm can obtain a symmetric loss $o(\epsilon)$ for this instance.

Then we show that a naive binary search algorithm achieves symmetric loss $O(\epsilon)$. The algorithm keeps track of a ***confidence interval*** $[\ell_t, r_t]$ that contains the true value v_t in each time step.

ALGORITHM 1: Symmetric loss minimizing algorithm for adversarial buyer with known changing rate ϵ

$\ell_1 \leftarrow 0$
$r_1 \leftarrow 1$
for each time step t **do**
    Price at $p_t \leftarrow \frac{r_t + \ell_t}{2}$
    if the current value $v_t < p_t$ **then**
        $\ell_{t+1} \leftarrow \max(0, \ell_t - \epsilon)$
        $r_{t+1} \leftarrow \min(1, p_t + \epsilon)$
    else
        $\ell_{t+1} \leftarrow \max(0, p_t - \epsilon)$
        $r_{t+1} \leftarrow \min(1, r_t + \epsilon)$
    end if
end for

At time t we have a confidence interval $[\ell_t, r_t] \ni v_t$ that contains the true value of the buyer, since the buyer's value only changes by at most ϵ. If the buyer can afford to pay $p_t = \frac{\ell_t + r_t}{2}$, i.e. $v_t \geq p_t$, it means that $v_{t+1} \in [p_t - \epsilon, r_t + \epsilon]$; otherwise, it means that $v_{t+1} \in [\ell_t - \epsilon, p_t + \epsilon]$. In both cases, we almost halve the length of the confidence interval. Let a_t be the length of the confidence interval at time t. Since the price is always the middle point of the interval, the symmetric loss of each time step is at most $\frac{a_t}{2}$. As a_t starts with 1 and almost halves in each step until $a_t = O(\epsilon)$, we have $\frac{1}{2} \sum_t a_t = O(T\epsilon)$, which upper bounds the total symmetric loss of all time steps in the algorithm.

$\square$

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 $a \leq 6\epsilon$

    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 $\leq 3\sqrt{\epsilon}$ 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 random-

ized 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 $v_{t_0} \geq \frac{1}{2}$. 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 $p_t \leq v_{t_0} - (t - t_0)\epsilon$ for each time t, the revenue loss of the algorithm when the values of the buyer increase in the phase is at least $\sum_{t_0 < t \leq t + \epsilon^{-1/2}} (v_{t_0} + (t - t_0)\epsilon - p_t) \geq \sum_{t_0 < t \leq t + \epsilon^{-1/2}} 2(t - t_0)\epsilon = \Omega(1)$.

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 $o(\epsilon^{1/2})$.

□

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 $\Omega(\epsilon^{2/3})$.*

Proof. We assume that $\epsilon \geq \frac{1}{T}$. Otherwise, we can run the algorithm with $\epsilon' = \frac{1}{T}$ and get revenue loss $(\epsilon')^{2/3} = o(1) = \tilde{O}(\epsilon^{2/3})$.

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 the length of each phase, while maintaining shorter confidence interval. The detailed algorithm is as follows.

ALGORITHM 5: Revenue loss minimizing algorithm for stochastic buyer with known changing rate ϵ

$\ell \leftarrow 0$
$r \leftarrow 1$
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

Now we analyze the revenue loss of the algorithm. In each phase $[t_0 + 1, t_0 + m]$ with $m = \epsilon^{-2/3}$, firstly the binary search algorithm is called with a revenue loss of $O(\log \frac{1}{\epsilon}) = \tilde{O}(1)$. Suppose that at time t_1 the search algorithm ends, and we get an estimate of v_{t_1} with additive accuracy 6ϵ. Let $\delta = 4\epsilon^{2/3}\sqrt{\log \frac{1}{\epsilon}}$. In the remaining of the phase, the price is fixed to be $\ell - \delta$. For any $t \in [t_1, t_0 + m]$, by Azuma's inequality, the probability that $v_t \notin [v_{t_1} - \delta, v_{t_1} + \delta]$ is

$$\Pr[v_t \notin [v_{t_1} - \delta, v_{t_1} + \delta]] \le 2 \exp\left(-\frac{\delta^2}{2m\epsilon^2}\right) < \frac{1}{m^2}. \tag{2}$$

By union bound, the probability that exists $t \in [t_1, t_0 + m]$ such that $v_t \notin [v_{t_1} - \delta, v_{t_1} + \delta]$ is at most $\frac{1}{m}$; in this case, the revenue loss in the phase is at most 1 per step thus m in total, so the expected revenue loss contributed from the case is $O(1)$. If $v_t \in [v_{t_1} - \delta, v_{t_1} + \delta]$ for every $t \in [t_1, t_0 + m]$, the revenue loss for setting price $\ell - \delta \ge v_t - 6\epsilon - \delta$ is at most $O(\epsilon + \delta)$ for each step, thus $mO(\epsilon + \delta) = O(\sqrt{\log \frac{1}{\epsilon}}) = \tilde{O}(1)$ in total. Combine both cases above, we know that the expected revenue loss in each phase is $\tilde{O}(1)$, thus $\tilde{O}(T\epsilon^{2/3})$ in total and $\tilde{O}(\epsilon^{2/3})$ revenue loss on average since there are $\frac{T}{m} = T\epsilon^{2/3}$ phases.

Now we show that such loss is tight, even when the buyer's values form an unbiased random walk with step size ϵ and starting value $\frac{1}{2}$.

Consider a game with multiple phases and the following more powerful seller. Each phase has at most $2m = 2\epsilon^{-2/3}$ time steps. At the beginning of each phase, the seller is told the exact value v_0 of the buyer at the current time. Then at each time step $t \in [1, 2m]$, the current value v_t increases or decreases by ϵ with equal probability, and the seller sets a price p_t. Whenever $p_t > v_t$, the phase terminates and incurs a loss $\ell = v_t$. If $p_t < v_t$ for the entire phase, the loss ℓ of the phase is defined by $\sum_{t \le 2m} \ell_t = \sum_{t \le 2m} (v_t - p_t)$. Then the game goes to the next phase, with v_0 reset to $v_{2m} \pm \epsilon$. The entire game ends when the total number of time steps in each phase sum up to T.

We notice that in the game, the seller is strictly more powerful, while the loss does not increase. In each phase before the phase terminates, the seller learns no information from the pricing. Thus without loss of generality, assume that given initial value v_0 of the buyer, the seller proposes a sequence of deterministic prices (randomness won't help) $p_1, p_2, \cdots, p_{2m}$, and then the values $v_1, \cdots, v_{2m}$ are realized randomly.

We do not count the loss contributed from the first m steps. Since Chernoff bound is tight up to constant factor for $U\{-1, 1\}$ random variables, i.e. $\Pr[v_t < v_0 - \epsilon\sqrt{t}] \ge \Omega(1)$ for $t \ge m$. Thus

8

for any $t \geq m$ if $p_t \geq v_0 - \epsilon\sqrt{m}$, with constant probability the phase stops at time t with loss v_t, which is $\Omega(1)$ in expectation. If $p_t < v_0 - \epsilon\sqrt{m}$, $\mathbb{E}[v_t - p_t | v_t \geq p_t] = \Omega(\epsilon\sqrt{m}) = \Omega(\epsilon^{2/3})$ since with probability $\frac{1}{2}$, $v_t \geq v_0$. Thus the loss of the phase is at least $m\Omega(\epsilon^{2/3}) = \Omega(1)$ in expectation if the phase ends at step $2m$.

The above analysis shows that at each phase, the expected loss is $\Omega(1)$. Since there are at least $\frac{T}{m} = T\epsilon^{2/3}$ phases, the total loss is at least $\Omega(T\epsilon^{2/3})$ in expectation. This finishes the proof of no randomized algorithm can get revenue loss $o(\epsilon^{2/3})$.

$\square$

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. The $\Omega(\epsilon)$ tightness result was shown in Theorem 4.1. Here we propose a pricing algorithm with a symmetric loss $\tilde{O}(\epsilon)$. 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 changing rate of the value exceeds $\hat{\epsilon}$.

ALGORITHM 6: 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**
         $\hat{\epsilon} \leftarrow \hat{\epsilon} \cdot 2$
         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

The algorithm deal 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 $\hat{\epsilon} > \epsilon$, $v_t \in [\ell_t, r_t]$ always

9

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 $\hat{\epsilon} > \epsilon$, then $v_t \geq p_t$ and $v_{t+1} < p_{t+1}$ will always hold. On the other hand, if $v_t \geq p_t$, then $v_{t+1} \geq p_t - \epsilon$, $v_{t+2} \geq p_t - 2\epsilon$ since the value of the buyer only changes at most ϵ additively, although the seller does not know the value of ϵ. Thus $v_t, v_{t+1}, v_{t+2} \geq \ell_t - 2\epsilon$. Similarly, if $v_{t+1} < p_{t+1} = r_t + \hat{\epsilon}$, then $v_t < p_{t+1} + \epsilon$, $v_{t+2} < p_{t+1} + \epsilon$. Thus $v_t, v_{t+1}, v_{t+2} < r_t + \epsilon + \hat{\epsilon}$. Since p_t, p_{t+1}, p_{t+2} are all in range $[\ell_t - 2\epsilon, r_t + \epsilon + \hat{\epsilon}]$, we have for any $j \in \{t, t+1, t+2\}$, $|v_j - p_j| = O(r_t - \ell_t + \epsilon + \hat{\epsilon})$. 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 proportional to the length of confidence bound plus the true changing rate and the estimated changing rate, i.e. $O(r_t - \ell_t + \epsilon + \hat{\epsilon})$.

To summarize, if in any three consecutive time steps the seller finds out $p_t > v_t$ or $p_{t+1} \leq v_{t+1}$, then $\hat{\epsilon} < \epsilon$ and $\hat{\epsilon}$ is doubled, and the symmetric loss obtained from time step $t, t+1, t+2$ is at most 1 each; if in any consecutive three time steps the seller observes $p_t \leq v_t$ and $p_{t+1} > v_{t+1}$, then the symmetric loss obtained from time step $t, t+1, t+2$ is $O(r_t - \ell_t + \epsilon + \hat{\epsilon})$ on average. Since $\hat{\epsilon}$ can only be doubled to at most 2ϵ, we know that there are $O(\log T)$ iterations with fixed $\hat{\epsilon}$. In each iteration, the time when $\hat{\epsilon}$ is doubled, the symmetric loss obtained from the previous three time steps is $O(1)$; in other cases, the symmetric loss obtained from the previous three time steps is constant times the length of the confidence interval plus $O(\epsilon + \hat{\epsilon})$. Since in each phase of fixed $\hat{\epsilon}$ the length of confidence interval starts with 1, and then decreases geometrically to $O(\hat{\epsilon})$, thus if the iteration contains a time steps, the total length of the confidence intervals is $O(1 + \hat{\epsilon}a) = O(1 + \epsilon a)$, and the total symmetric loss of the phase is also $O(1 + \epsilon a)$. Thus the total symmetric loss of the algorithm is $O(\log T + T\epsilon) = \tilde{O}(T\epsilon)$, which is $\tilde{O}(\epsilon)$ on average. $\qquad \square$

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 6 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 7: Revenue loss minimizing algorithm for adversarial buyer with unknown changing rate ϵ

$\hat{\epsilon} \leftarrow \frac{1}{T}$
while $\hat{\epsilon} < \frac{1}{2}$ **do**
    for each phase $[t_0 + 1, t_0 + m_{\hat{\epsilon}}]$ of length $m_{\hat{\epsilon}} = \hat{\epsilon}^{-1/2}$ **do**
        Apply Algorithm 2 to locate the current value of the buyer in an interval $[\ell_t, r_t]$ with length $\sqrt{\hat{\epsilon}}$
        Randomly select a $t^* \in [t_0 + 1, t_0 + m_{\hat{\epsilon}}]$
        for each step t in the phase **do**
            if $t \neq t^*$ **then**
                Price at $p_t \leftarrow \ell_t$
                if $v_t < p_t$ **then**
                    $\hat{\epsilon} \leftarrow 2\hat{\epsilon}$ and break (terminate the phase)
                end if
            else
                Price at $p_t \leftarrow r_t$
                if $v_t \geq p_t$ **then**
                    $\hat{\epsilon} \leftarrow 2\hat{\epsilon}$ and break (terminate the phase)
                end if
            end if
            $[\ell_{t+1}, r_{t+1}] \leftarrow [\ell_t - \epsilon, r_t + \epsilon]$
        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 7 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 7 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 $\Omega(\epsilon^{2/3})$.*

Proof. The $\Omega(\epsilon^{2/3})$ tightness of the theorem follows from Theorem 4.3. Now we provide an algorithm with revenue loss $\tilde{O}(\epsilon^{2/3})$. The algorithm combines Algorithm 4 for stochastic buyer with known changing rate and Algorithm 7 for adversarial buyer with unknown changing rate as follows. For every iteration with guess $\hat{\epsilon}$ in Algorithm 7, replace each phase of length $\hat{\epsilon}^{-1/2}$ by a phase in Algorithm 4 with a random checking step that prices at the top of the confidence interval which tries to find the evidence of $\hat{\epsilon} < \epsilon$. The analysis is almost the same as the analysis for Algorithm 7.

ALGORITHM 8: Revenue loss minimizing algorithm for stochastic buyer with unknown changing rate ϵ

$\hat{\epsilon} \leftarrow \frac{1}{T}$
while $\hat{\epsilon} < \frac{1}{2}$ **do**
    for each phase $[t_0 + 1, t_0 + m_{\hat{\epsilon}}]$ of length $m_{\hat{\epsilon}} = \hat{\epsilon}^{-2/3}$ **do**
        Apply Algorithm 3 to locate the current value of the buyer in an interval $[\ell, r]$ with length $\sqrt{\hat{\epsilon}}$
        Randomly select a $t^* \in [t_0 + 1, t_0 + m_{\hat{\epsilon}}]$
        for each step t in the phase **do**
            if $t \neq t^*$ **then**
                Price at $p_t \leftarrow \ell - 4\hat{\epsilon}^{-2/3}\sqrt{\log T}$
                if $v_t < p_t$ **then**
                    $\hat{\epsilon} \leftarrow 2\hat{\epsilon}$
                    break (terminate the phase)
                end if
            else
                Price at $p_t \leftarrow r + 4\hat{\epsilon}^{-2/3}\sqrt{\log T}$
                if $v_t \geq p_t$ **then**
                    $\hat{\epsilon} \leftarrow 2\hat{\epsilon}$
                    break (terminate the phase)
                end if
            end if
        end for
    end for
end while

Now we analyze the revenue loss of the algorithm. In each iteration with changing rate estimate $\hat{\epsilon}$, let $\delta = 4\hat{\epsilon}^{-2/3}\sqrt{\log T}$. If $\hat{\epsilon} \geq \epsilon$, then the probability that $v_t \notin [\ell - \delta, r + \delta]$ is at most $\frac{1}{T^2}$ for each step t (as analyzed in (2)). By union bound, the probability that the algorithm falsely doubles $\hat{\epsilon}$ when $\hat{\epsilon} \geq \epsilon$ is at most $\frac{1}{T}$ in all time steps. Therefore the revenue loss contributed from this case is $O(1)$, since the total revenue loss when this case happens is at most T.

Then we assume that $\hat{\epsilon}$ is always at most ϵ, and the rest of the analysis is identical to the analyze of Algorithm 7. In each iteration with fixed $\hat{\epsilon}$, when a bad event $v_t \notin [\ell - \delta, r + \delta]$ is detected, in expectation $m_{\hat{\epsilon}} = \hat{\epsilon}^{-2/3}$ bad events have occurred, thus the revenue loss contributed from bad events is $\hat{\epsilon}^{-2/3}$ in an iteration with changing rate estimate $\hat{\epsilon}$, thus $\sum_{\hat{\epsilon}} \hat{\epsilon}^{-2/3} = O(T^{2/3})$ in total. The revenue loss for each time step with good event $v_t \notin [\ell - \delta, r + \delta]$ is $O(\delta) = \tilde{O}(\hat{\epsilon}^{2/3})$, thus $\tilde{O}(T\hat{\epsilon}^{2/3}) = \tilde{O}(T\epsilon^{2/3})$ in total. Combining the revenue loss of steps with good events, steps with bad events, and cases where $\hat{\epsilon} > \epsilon$, we have the total revenue loss of the algorithm for all time steps is $\tilde{O}(T\epsilon^{2/3}) + O(T^{2/3}) + \tilde{O}(1) = T\tilde{O}(\epsilon^{2/3})$.

Note: The current revenue loss of the algorithm is actually $O(\epsilon^{2/3} \log T)$. It can get improved to $O(\epsilon^{2/3} polylog \frac{1}{\epsilon})$, if the algorithm sets $\delta = 4\hat{\epsilon}^{2/3} \log^4 \frac{1}{\hat{\epsilon}}$, and doubles $\hat{\epsilon}$ whenever the current total number of bad events detected for the current $\hat{\epsilon}$ exceeds $\frac{1}{m^2}$ fraction of the current total number of time steps. By Azuma's inequality, we can argue that the probability that $\hat{\epsilon} > \epsilon$ is falsely doubled is less than $\frac{1}{m^2 t^2}$ in step t, which will only in expectation contribute to revenue loss $\frac{T}{m^2} = O(T\hat{\epsilon}^{2/3})$ in total. The revenue loss from good events is $O(\delta) = \tilde{O}(\hat{\epsilon}^{2/3})$ per step. Thus the average revenue loss is $\tilde{O}(\epsilon^{2/3})$ per step. $\qquad \square$

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.

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 $\Omega(\bar{\epsilon})$.*

Proof. The tightness result $\Omega(\bar{\epsilon})$ was shown in Theorem 4.1. Now we propose an algorithm with symmetric loss $\tilde{O}(\bar{\epsilon})$.

We propose the following algorithm for the seller that repeatedly guess 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 reduce the value of the guess $\hat{\epsilon}$ by a factor of $\frac{1}{2}$ if in $O(\log T)$ time steps the algorithm cannot find any evidence supporting $\hat{\epsilon} < \epsilon_t$. Whenever the algorithm finds a piece of evidence that $\hat{\epsilon} < \epsilon_t$, 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 symmetric loss bounded. Below we show the algorithm in detail.

The same as Algorithm 6 for unknown fixed changing rate, the algorithm takes a round of 3 time steps each time, and always keeps a confidence interval $[\ell_t, r_t]$ for the true value of the buyer at each time step. The same as in Algorithm 1, for three consecutive time steps $3k+1, 3k+2, 3k+3$ in round k, the algorithm first prices at the bottom of the confidence interval, then at the top of the confidence interval, finally in the middle of the interval. If $\hat{\epsilon} \geq \epsilon_{3k+1}$, this is going to be a "good round" with $v_{3k+1} \geq p_{3k+1}$ and $v_{3k+2} < p_{3k+2}$, so no bad evidence is found by the algorithm. If this holds for $\log^3 T$ consecutive rounds, it means that the algorithm has been stable with the current $\hat{\epsilon}$, and the confidence interval has been $O(\hat{\epsilon}_{3k+1})$ for $\Omega(\log^3 T)$ steps. Then the algorithm decreases $\hat{\epsilon}$ by a factor of $\frac{1}{2}$ to try to explore the possibility that the true changing rate ϵ_{3k+1} is

ALGORITHM 9: Symmetric loss minimizing algorithm for adversarial buyer with unknown decreasing changing rate ϵ_t

---

Set confidence interval $[\ell_1, r_1] \leftarrow [0, 1]$
Set changing rate estimate $\hat{\epsilon} \leftarrow \frac{1}{2}$
Let $a \leftarrow 0$ be the recent round without bad evidence
Let $s \leftarrow 0$ be the number of rounds for the current $\hat{\epsilon}$
for each round k with three consecutive time steps $3k+1, 3k+2, 3k+3$ **do**
    $s \leftarrow s + 1$
    Set price $p_{3k+1} \leftarrow \ell_{3k+1}$
    Set price $p_{3k+2} \leftarrow r_{3k+2} = r_{3k+1} + \hat{\epsilon}$
    Set price $p_{3k+3} \leftarrow \frac{\ell_{3k+1} + r_{3k+1}}{2}$
    if the seller finds $v_{3k+1} \geq \ell_{3k+1}$ and $v_{3k+2} < r_{3k+2}$ **then**
        $a \leftarrow k$
        if $v_{3k+3} < p_{3k+3}$ **then**
            $[\ell_{3k+4}, r_{3k+4}] \leftarrow [\ell_{3k+1} - 3\hat{\epsilon}, p_{3k+3} + \hat{\epsilon}]$
        else
            $[\ell_{3k+4}, r_{3k+4}] \leftarrow [p_{3k+3} - \hat{\epsilon}, r_{3k+1} + 3\hat{\epsilon}]$
        end if
        if $s > \log^3 T$ and $\hat{\epsilon} > \frac{1}{T}$ **then**
            $\hat{\epsilon} \leftarrow \hat{\epsilon}/2$
        end if
    else
        $\hat{\epsilon} \leftarrow 2\hat{\epsilon}$
        $[\ell_{3k+4}, r_{3k+4}] \leftarrow [\ell_{3a+1} - (3k+3-3a)\hat{\epsilon}, r_{3a+1} + (3k+3-3a)\hat{\epsilon}]$
    end if
end for

---

currently much smaller than $\hat{\epsilon}$. However, when $\hat{\epsilon} < \epsilon_{3k+1}$, the confidence interval may expand not enough to contain the true value of the buyer, so there is possibility that this round becomes a "bad round" with $v_{3k+1} < p_{3k+1}$ or $v_{3k+2} \geq p_{3k+2}$. In this case, the algorithm doubles $\hat{\epsilon}$. The confidence bound is not accurate as $\hat{\epsilon} < \epsilon_{3k+1}$ previously. However, if the new $\hat{\epsilon} \geq \epsilon_{3k+1}$, the algorithm gets the correct confidence bound $[\ell_{3a+1}, r_{3a+1}]$ in the previous good round a, and calculates a correct confidence bound $[\ell_{3k+4}, r_{3k+4}] \leftarrow [\ell_{3a+1} - (3k+3-3a)\hat{\epsilon}, r_{3a+1} + (3k+3-3a)\hat{\epsilon}]$ for time step $3k+4$, since the buyer's value cannot change more than $(3k+3-3a)\epsilon_{3a+1} < (3k+3-3a)\hat{\epsilon}$ in $3k+3-3a$ steps.

Now we analyze the performance of the algorithm. Partition time horizon $[T]$ to $\log T$ intervals $I_1, I_2, \cdots, I_{\log T}$, such that for each time interval I_i and time $t \in I_i$, $\epsilon_t \in (2^{-i}, 2^{-i+1}]$. In other words, each time interval contains time steps with similar changing rate. For any time interval I_i, let $\epsilon_i^* = 2^{-i+1}$. We argue that in time interval I_i the symmetric loss is $O(\log^4 T + |I_i|\epsilon_i^* = O(\log^4 T + \sum_{t \in I_i} \epsilon_t)$, and the theorem follows immediately by taking the average of the symmetric loss in all time intervals.

Notice that in I_i, the algorithm may start with $\hat{\epsilon} > \epsilon_i^*$, then decrease gradually to reach $\hat{\epsilon} = \epsilon_i^*$. The symmetric loss in this process is $O(\log^4 T)$, since there will be no bad rounds, and $\hat{\epsilon}$ will stay at each $\hat{\epsilon} > \epsilon_i^*$ for at most $O(\log^3 T)$ steps, thus reduce to ϵ_i^* in $O(\log^4 T)$ steps. Now we analyze what happens when $\hat{\epsilon} \leq \epsilon_i^*$.

For any bad round k, since it is at most b rounds from the closest good round $a = k - b$ with confidence interval $[\ell_{3a+1}, r_{3a+1}]$, all of the values and prices in the round are bounded by interval

$[\ell_{3a+1} - 3b\epsilon_i^*, r_{3a+1} + 3b\epsilon_i^*] \subseteq [\ell_{3a+1} - 3\log T\epsilon_i^*, r_{3a+1} + 3\log T\epsilon_i^*]$, since in round a all of the values and confidence boundaries are bounded by $[\ell_{3a+1} - O(1)\epsilon_i^*, r_{3a+1} + O(1)\epsilon_i^*]$. Thus the symmetric loss for any bad round that is b rounds away from a good round is $O(r_{3a+1} - \ell_{3a+1}) + O(\epsilon_i^*) + 6\log T\epsilon_i^* = O(\log T\epsilon_i^*)$.

As analyzed in Algorithm 6 for static changing rate, in a good round k without bad evidence, the symmetric loss in the round is proportional the length of the confidence interval plus the true changing rate ϵ_{3k+1} and the estimated changing rate $\hat{\epsilon}_k$ for the round, i.e. $O(r_{3k+1} - \ell_{3k+1} + \hat{\epsilon}_k + \epsilon_{3k+1})$. In such a good round, since $\hat{\epsilon}_k$, ϵ_{3k+1} and the length of confidence interval $r_{3k+1} - \ell_{3k+1}$ are all $O(\epsilon_i^*)$, the good round has symmetric loss $O(\epsilon_i^*) \leq c\epsilon_i^*$ for some constant c.

The same as in the analysis of Algorithm 10 for the revenue loss setting, we can use the loss of good rounds to compensate the loss of the bad rounds. We argue that the per-step symmetric loss in I_i is $\leq 2c\epsilon_i^*$. Between two blocks of consecutive bad rounds, there are at least $O(\log^3 T)$ good rounds. In a block of bad rounds, the symmetric loss is $O(\log T\epsilon_i^*)$ more than the expected $2c\epsilon_i^*$ loss per step, thus $O(\log^2 T\epsilon_i^*)$ more in total in the block of bad rounds. At the same time, In the $O(\log^3 T)$ consecutive good rounds, the symmetric loss is $c\epsilon_i^*$ less than the expected $2c\epsilon_i^*$ loss per step, thus $O(\log^3 T\epsilon_i^*)$ less in total in the block of good rounds.

Combine the cases above for good rounds and bad rounds, we know that in time interval I_i with $\hat{\epsilon} \leq \epsilon_i^*$, the symmetric loss is $\leq c\epsilon_i^*$ in each round on average. This finishes the proof of the total symmetric loss in the time interval being $O(\log^4 T + |I_i|\epsilon_i^* = O(\log^4 T + \sum_{t \in I_i} \epsilon_t)$.

Thus the total symmetric loss of the algorithm is $\sum_{i \leq \log T} O(\log^4 T + \sum_{t \in I_i} \epsilon_t) = To(1) + O(1)\sum_{t \in I_i} \epsilon_t = T\tilde{O}(\bar{\epsilon})$.

□

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 7, and will halve $\hat{\epsilon}$ when the buyer passes all checking steps. The algorithm is described in Algorithm 10.

ALGORITHM 10: Revenue loss minimizing algorithm for adversarial buyer with unknown decreasing changing rate ϵ_t

---

$\hat{\epsilon} \leftarrow \frac{1}{2}$
while true **do**
   for $\hat{\epsilon}^{-1/2}$ phases of length $m_{\hat{\epsilon}} = \hat{\epsilon}^{-1/2}$ **do**
      At the beginning of phase $[t_0 + 1, t_0 + m_{\hat{\epsilon}}]$, apply Algorithm 2 to locate the current value of the
      buyer in an interval $[\ell_t, r_t]$ with length $\sqrt{\hat{\epsilon}}$
      Randomly select a $t^* \in [t_0 + 1, t_0 + m_{\hat{\epsilon}}]$
      for each step t in the phase **do**
         if $t \neq t^*$ **then**
            Price at $p_t \leftarrow \ell_t$
            if $v_t < p_t$ **then**
               $\hat{\epsilon} \leftarrow 2\hat{\epsilon}$ and go back to the beginning of the while loop (terminate the $\hat{\epsilon}^{-1/2}$ phases)
            end if
         else
            Price at $p_t \leftarrow r_t$
            if $v_t \geq p_t$ **then**
               $\hat{\epsilon} \leftarrow 2\hat{\epsilon}$ and go back to the beginning of the while loop (terminate the $\hat{\epsilon}^{-1/2}$ phases)
            end if
         end if
         $[\ell_{t+1}, r_{t+1}] \leftarrow [\ell_t - \epsilon, r_t + \epsilon]$
      end for
   end for
   $\hat{\epsilon} \leftarrow \hat{\epsilon}/2$ if $\hat{\epsilon} > \frac{1}{T}$
end while

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 found, as shown in the analysis of Algorithm 7, 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

16

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

$$\sum_{i \leq \log T} \tilde{O}((\epsilon_i^*)^{-1/2} + |I_i|(\epsilon_i^*)^{1/2})$$

$$= \tilde{O}(T^{1/2}) + \tilde{O}(1) \sum_i |I_i|(\epsilon_i^*)^{1/2}$$

$$\leq To(1) + \tilde{O}(1) \sum_i |I_i|\bar{\epsilon}^{1/2} = \tilde{O}(T\bar{\epsilon}^{1/2}),$$

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

$\square$

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 $\Omega(\bar{\epsilon}^{2/3})$.*

Proof. The $\Omega(\bar{\epsilon}^{2/3})$ tightness follows from Theorem 4.3. The algorithm for $\tilde{O}(\bar{\epsilon}^{2/3})$ revenue loss can be obtained by combining Algorithm 8 for stochastic buyer with unknown fixed changing rate, and Algorithm 10 for adversarial buyer with unknown decreasing changing rate. In particular, we replace each iteration of fixed $\hat{\epsilon}$ with $\hat{\epsilon}^{-1/2}$ phases of length $\hat{\epsilon}^{-1/2}$, by $\hat{\epsilon}^{-2/3}$ phases of length $\hat{\epsilon}^{-2/3}$ in Algorithm 8.

The analysis of the algorithm is identical to the analysis of Algorithm 10. Partition the time horizon $[T]$ to intervals $I_1, \cdots, I_{\log T/2}$, each with $\epsilon_t \in (2^{-i}, 2^{-i+1} = \epsilon_i^*]$ for any $t \in I_i$. We can prove that the total revenue loss in each time interval is $\tilde{O}((\epsilon_i^*)^{-2/3} + |I_i|(\epsilon_i^*)^{2/3})$. The total revenue loss of the algorithm is

$$\sum_{i \leq \log T} \tilde{O}((\epsilon_i^*)^{-2/3} + |I_i|(\epsilon_i^*)^{2/3})$$

$$= \tilde{O}(T^{2/3}) + \tilde{O}(1) \sum_i |I_i|(\epsilon_i^*)^{2/3}$$

$$\leq To(1) + \tilde{O}(1) \sum_i |I_i| \left(\frac{\sum_i |I_i|\epsilon_i^*}{\sum_i |I_i|} \right)^{2/3}$$

$$= \tilde{O}(T\bar{\epsilon}^{2/3}),$$

Here the inequality follows from Hölder's inequality and $\bar{\epsilon} = \frac{1}{T} \sum_t \epsilon_t$. Thus the average revenue loss of each time step is $\tilde{O}(\bar{\epsilon}^{2/3})$.

$\square$

17

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. 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 $\epsilon_t \geq |v_{t+1} - v_t|$. 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 $[\frac{\ell_t + r_t}{2}, r_t + \delta_j]$. Let $a_t = r_t - \ell_t$ be the length of the confidence interval at time t, then $a_{t+1} = \frac{1}{2}a_t + \delta_j$. Since v_{t+1} and all prices at this time step are in $[\ell_t - \delta_j, r_t + \delta_j]$, thus the symmetric loss of each query is at most $2a_t + 2\delta_j = 4a_{t+1}$. Also notice that $v_{t+1} \notin [\ell_t - \delta_j/2, r_t + \delta_j/2]$ since the algorithm does not stop at $j - 1$ at time t, thus $\epsilon_t > \frac{\delta_j}{2}$ as the buyer's value must has changed by $\frac{\delta_j}{2}$ at this step. Therefore the symmetric loss of each time step $t + 1$ is at most $4a_{t+1}$, while $a_{t+1} \leq \frac{1}{2}a_t + \epsilon_t$. Then the total symmetric loss of the algorithm is $\leq 4\sum_{t\in[T]} a_t \leq 8\sum_{t\in[T]} \epsilon_t$, which implies the average symmetric loss is $O(\bar{\epsilon})$.

However, we are not allowed to have multiple pricing queries for the same value. The key observation to be proved in this section 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 above algorithm has at most $O(\log T) = \tilde{O}(1)$ pricing queries in each step, the serialized algorithm's symmetric loss only increases by $\tilde{O}(1)$.

The serialized algorithm runs as follows. The algorithm runs in phases, with each phase i having time steps $[t_i + 1, t_{i+1}]$ for a to-be-determined stopping time t_{i+1}. In each phase, the algorithm starts with an estimated confidence interval $[\ell_{t_i+1}, r_{t_i+1}]$, and repeatedly set prices $\ell_{t_i+1} - \delta_j$ and $r_{t_i+1} + \delta_j$ for every two time steps. If the algorithm observes $v_{t_i+2j+1} \geq \ell_{t_i+1} - \delta_j$ and $v_{t_i+2j+2} < r_{t_i+1} + \delta_j$, the phase stops at time $t_{i+1} = t_i + 2j + 3$ with price $p_{t_{i+1}} = \frac{\ell_{t_i+1}+r_{t_i+1}}{2}$ trying to halve the length of the confidence interval.

Now we analyze the symmetric loss of each phase $I_i = [t_i + 1, t_{i+1}]$. The first observation is that at each time step, the true value may not be in the confidence interval, due to the delay of prices. The confidence interval at time $t_i + 1$ is $[\ell_{t_i+1}, r_{t_i+1}]$ in the algorithm, but since the value of ℓ_{t_i+1} and r_{t_i+1} are determined in the previous three time steps, thus the "true" confidence interval can be defined by $v_{t_i+1} \in [\ell_{t_i+1} - \epsilon_{t_i-2} - \epsilon_{t_i-1} - \epsilon_{t_i}, r_{t_i+1} + \epsilon_{t_i-2} + \epsilon_{t_i-1} + \epsilon_{t_i}]$. Furthermore, for any time $t' \in I_i$, $v_{t'} \in [\ell_{t_i+1} - \sum_{t_i-2\leq t<t_{i+1}} \epsilon_t, r_{t_i+1} + \sum_{t_i-2\leq t<t_{i+1}} \epsilon_t]$.

Observe that the algorithm in phase i breaks at j, thus $v_{t_i+2j+1} \geq p_{t_i+2j+1} = \ell_{t_i+1} - \delta_j$ and $v_{t_i+2j+2} < p_{t_i+2j+2} = r_{t_i+1} + \delta_j$. Thus any price in the phase is in range $[\ell_{t_i+1} - \delta_j, \ell_{t_i+1} + \delta_j]$, while any value v_t in the phase is in range $[v_{t_i+2j+1} - \sum_{t\in I_i} \epsilon_t, v_{t_i+2j+2} + \sum_{t\in I_i} \epsilon_t] \subseteq [\ell_{t_i+1} - \delta_j - \sum_{t\in I_i} \epsilon_t, r_{t_i+1} + \delta_j + \sum_{t\in I_i} \epsilon_t]$, so the symmetric loss at any time step in the phase is at most $r_{t_i+1} - \ell_{t_i+1} + 2\delta_j + \sum_{t\in I_i} \epsilon_t$.

18

ALGORITHM 11: Symmetric loss minimizing algorithm for adversarial buyer with unknown dynamic changing rate ϵ_t

---

$[\ell_1, r_1] \leftarrow [0, 1]$

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 two consecutive time steps $t_i + 2j + 1, t_i + 2j + 2$ with $j \geq 0$ **do**

        $\hat{\epsilon} \leftarrow \delta_j = 2^j T^{-1}$

        Set price $p_{t_i+2j+1} \leftarrow \ell_{t_i+1} - \delta_j$

        Set price $p_{t_i+2j+2} \leftarrow r_{t_i+1} + \delta_j$

        if the seller finds $v_{t_i+2j+1} \geq p_{t_i+2j+1}$ and $v_{t_i+2j+2} < p_{t_i+2j+2}$ **then**

            break (from this for loop)

        end if

    end for

    $t_{i+1} \leftarrow t_i + 2j + 3$

    Set $p_{t_{i+1}} \leftarrow \frac{\ell_{t_i+1} + r_{t_i+1}}{2}$

    if $v_{t_{i+1}} < p_{t_{i+1}}$ **then**

        $[\ell_{t_{i+1}+1}, r_{t_{i+1}+1}] \leftarrow [\ell_{t_i+1} - \delta_j, p_{t_{i+1}}]$

    else

        $[\ell_{t_{i+1}+1}, r_{t_{i+1}+1}] \leftarrow [p_{t_{i+1}}, r_{t_i+1} + \delta_j]$

    end if

end for

Also since the algorithm does not break at $j-1$ in the for loop, thus either $v_{t_i+2j-1} < p_{t_i+2j-1} = \ell_{t_i+1} - \delta_{j-1}$, or $v_{t_i+2j} > p_{t_i+2j} = r_{t_i+1} + \delta_{j-1}$. In either case, $\sum_{t_i-2 \leq t < t_{i+1}} \epsilon_t > \delta_{j-1} = \frac{1}{2}\delta_j$.

Let $a_i = r_{t_i+1} - \ell_{t_i+1}$ be the length of the confidence interval at the beginning of phase i, and $b_i = \sum_{t \in I_i} \epsilon_t$ be the total changing rate of the value in phase i. Then according to the binary search step, $a_{i+1} = \frac{1}{2}a_i + \delta_{j_i}$ where j_i is the value of j at the end of phase i. Since $\delta_{j_i} = 2\delta_{j_i-1} < \sum_{t_i-2 \leq t < t_{i+1}} \epsilon_t < b_{i-1} + b_i$, we have

$$a_{i+1} < \frac{1}{2}a_i + 2(b_i + b_{i-1}).$$

Then a_{i+1} can be represented using only b_i's as follows:

$$a_{i+1} < 2(b_i + b_{i-1}) + (b_{i-1} + b_{i-2}) + \frac{1}{2}(b_{i-2} + b_{i-3}) + \cdots \tag{3}$$

Since the total symmetric loss in phase i is upper bounded by $a_{i+1} + \sum_{t \in I_i} \epsilon_t$ per step thus $|I_i|(a_{i+1} + b_i) \leq (2 \log T + 1)(a_{i+1} + b_i)$ in total, summing over all phases we get the total symmetric loss of the algorithm for all phases is $L := (2 \log T + 1) \sum_i (a_{i+1} + b_i)$. L can be represented with only b_i by applying (3) to L. Since the coefficient of b_i in L is $O(\log T)$ for each b_i, thus $L = O(\log T) \sum_i b_i = O(\log T) \sum_{t \in [T]} \epsilon_t$. Therefore the average symmetric loss of the algorithm is $\tilde{O}(\bar{\epsilon})$.

$\square$

References

[1] Kareem Amin, Afshin Rostamizadeh, and Umar Syed. 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*, pages 622–630, 2014. 1

[2] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Proceedings of IEEE 36th Annual Foundations of Computer Science*, pages 322–331. IEEE, 1995. 21

[3] Nicolo Cesa-Bianchi, Tommaso Cesari, and Vianney Perchet. Dynamic pricing with finitely many unknown valuations. In *Algorithmic Learning Theory*, pages 247–273. PMLR, 2019. 1

[4] Shuchi Chawla, Nikhil R. Devanur, Anna R. Karlin, and Balasubramanian Sivan. Simple pricing schemes for consumers with evolving values. In Robert Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1476–1490. SIAM, 2016. 2

[5] Maxime C Cohen, Ilan Lobel, and Renato Paes Leme. Feature-based dynamic pricing. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, pages 817–817. ACM, 2016. 1

[6] Amit Daniely, Alon Gonen, and Shai Shalev-Shwartz. Strongly adaptive online learning. In *International Conference on Machine Learning*, pages 1405–1411. PMLR, 2015. 2

[7] Nikhil R. Devanur, Yuval Peres, and Balasubramanian Sivan. Perfect bayesian equilibria in repeated sales. *Games Econ. Behav.*, 118:570–588, 2019. 1

[8] Dylan J Foster, Zhiyuan Li, Thodoris Lykouris, Karthik Sridharan, and Éva Tardos. Learning in games: robustness of fast convergence. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 4734–4742, 2016. 2

[9] Elad Hazan and Comandur Seshadhri. Adaptive algorithms for online decision problems. In *Electronic colloquium on computational complexity (ECCC)*, volume 14, 2007. 2

[10] Mark Herbster and Manfred K Warmuth. Tracking the best linear predictor. *Journal of Machine Learning Research*, 1(281-309):10–1162, 2001. 2

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

[12] Sham M. Kakade, Ilan Lobel, and Hamid Nazerzadeh. Optimal dynamic mechanism design and the virtual-pivot mechanism. *Oper. Res.*, 61(4):837–854, 2013. 2

[13] Robert Kleinberg and Tom Leighton. 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*, pages 594–605. IEEE, 2003. 1, 4, 21, 22

[14] Akshay Krishnamurthy, Thodoris Lykouris, Chara Podimata, and Robert Schapire. Contextual search in the presence of irrational agents. *arXiv preprint arXiv:2002.11650*, 2020. 1

[15] Renato Paes Leme and Jon Schneider. Contextual search via intrinsic volumes. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 268–282, 2018. doi: 10.1109/FOCS.2018.00034. URL https://doi.org/10.1109/FOCS.2018.00034. 1

[16] Allen Liu, Renato Paes Leme, and Jon Schneider. Optimal contextual pricing and extensions. pages 1059–1078, 2021. doi: 10.1137/1.9781611976465.66. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611976465.66. 1

[17] Ilan Lobel, Renato Paes Leme, and Adrian Vladu. Multidimensional binary search for contextual decision-making. *Operations Research*, 2017. 1

[18] Haipeng Luo and Robert E Schapire. Achieving all with no parameters: Adanormalhedge. In *Conference on Learning Theory*, pages 1286–1304. PMLR, 2015. 2

[19] Thodoris Lykouris, Karthik Sridharan, and Éva Tardos. Small-loss bounds for online learning with partial information. In *Conference on Learning Theory*, pages 979–986. PMLR, 2018. 2

[20] Renato Paes Leme, Balasubramanian Sivan, and Yifeng Teng. Why do competitive markets converge to first-price auctions? In *Proceedings of The Web Conference 2020*, pages 596–605, 2020. 1

[21] Alessandro Pavan, Ilya Segal, and Juuso Toikka. 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. 2

[22] Virag Shah, Ramesh Johari, and Jose Blanchet. Semi-parametric dynamic contextual pricing. 32:2363–2373, 2019. URL https://proceedings.neurips.cc/paper/2019/file/363763e5c3dc3a68b399058c34aecf2c-Paper.pdf. 1

[23] Steven E Shreve. *Stochastic calculus for finance II: Continuous-time models*, volume 11. Springer Science & Business Media, 2004. 2

A   Comparison to Fixed Price Benchmark

Theorem A.1. *The no-regret algorithm for an adversarial buyer in [13] can have $\Omega(1)$ revenue loss.*

Proof. Consider the following sequence of buyer values $\mathbf{v} = (v_1, \cdots, v_T)$ with changing rate ϵ. Let $m = \frac{1}{\epsilon}$. For any integer $k \geq 0$ and $1 \leq j \leq m$, let $v_{2km+j} = j\epsilon$, and $v_{2km+m+j} = 1 - (j-1)\epsilon$. In other words, the value sequence is periodic with period $2m$; in each phase of repetition, v_t starts with ϵ and increases by ϵ for each step until it reaches 1, then starts with 1 and decreases by ϵ for each step.

We show that the zero-regret pricing algorithm in [13] gives $\Omega(T)$ revenue loss, if $T > m^3 \ln m = \frac{1}{\epsilon^3} \ln \frac{1}{\epsilon}$. The algorithm applies the seminal bandit algorithm EXP3 [2] with each arm $1 \leq i \leq m$

21

corresponding to a price $p_i = i\epsilon$ which is a multiple of ϵ. We describe the algorithm in detail as follows.

ALGORITHM 12: No-regret pricing algorithm in [13]

$\eta \leftarrow \sqrt{\frac{\ln m}{Tm}}$

$w_1(i) \leftarrow 1$ for each arm $1 \leq i \leq m$

for each time step t **do**

    $q_t(i) \leftarrow (1 - \eta)\frac{w_t(i)}{W_t} + \frac{\eta}{m}$

    Draw i_t from $\mathbf{q}_t$, i.e. $\Pr[i_t = i] = q_t(i)$ for each $1 \leq i \leq m$

    Price at $p_t \leftarrow i_t\epsilon$, and observe revenue gain $r_t \leftarrow p_t\mathbf{1}\{v_t \geq p_t\}$

    Update $w_{t+1}(i) \leftarrow w_t(i)$ for $i \neq i_t$, and $w_{t+1}(i_t) \leftarrow w_t(i_t)\exp\left(\eta\frac{r_t}{mq_t(i_t)}\right)$

end for

It suffices to show that the revenue loss in each time interval $I = [2mk + 1, 2mk + m]$ is $\Omega(m)$ in expectation. Partition the arms to two groups $S_1 = \{1, 2, \cdots, \frac{m}{2}\}$ and $S_2 = \{\frac{m}{2} + 1, \cdots, m\}$. Then we have the following observations.

Observation 1: with probability $> \frac{1}{e}$, $q_t(i_t) \geq \frac{1}{m^2}$ for each time step $t \in I$. This is because the probability of selecting an arm i with $q_t(i) < \frac{1}{m^2}$ is at most $\frac{m-1}{m^2}$ since there are at most $m - 1$ such arms, thus the probability of $q_t(i_t) \geq \frac{1}{m^2}$ for each $t \in I$ is at least $(1 - \frac{m-1}{m^2})^m > \frac{1}{e}$.

Observation 2: with probability $> \frac{1}{e}$, $w_t(i)$ increases to a factor of e in the entire time interval I for every arm $1 \leq i \leq m$. This is because if $q_t(i_t) \geq \frac{1}{m^2}$, $w_{t+1}(i_t)$ increases to a fraction of $\exp\left(\eta\frac{r_t}{mq_t(i_t)}\right) \leq \exp\left(\sqrt{\frac{\ln m}{Tm}}\frac{1}{m \cdot m^{-2}}\right) < e^{1/m}$ compared to $w_t(i_t)$ for each time step t since $T > m^3 \ln m$. Thus if $q_t(i_t) \geq \frac{1}{m^2}$ for each time step $t \in I$, then for any arm $1 \leq i \leq m$ and time step $t \in I$, $w_t(i) < ew_{2mk+1}(i)$ as there are only m steps in I. Then Observation 2 follows from Observation 1.

Then we are ready to argue that in time interval I, the total revenue loss of Algorithm 12 is $\Omega(m)$. We prove a stronger result, that the algorithm has $\Omega(m)$ revenue loss in either time interval $I_1 = [2mk + \frac{m}{4} + 1, 2mk + \frac{m}{2}]$ or $I_2 = [2mk + \frac{3m}{4} + 1, 2mk + m]$. Let $W_{t,1} = \sum_{i \in S_1} w_t(i)$ be the total weight of arms in $S_1 = \{1, 2, \cdots, \frac{m}{2}\}$ at time step $t \in I$, and similarly define $W_{t,2} = \sum_{i \in S_2} w_t(i)$ for arms S_2. If $W_{2mk+1,1} \geq W_{2mk+1,2}$ at the beginning of I, by Observation 2 $W_{t,1} \geq \frac{1}{e}W_{t,2}$ for any time step $t \in I$. By definition of $q_t(i)$, with constant probability $> \frac{1/e}{1+1/e} = \frac{1}{e+1}$, the algorithm selects an arm $i \in S_1$ at any time step $t \in I$. In particular, with probability $\frac{1}{e+1}$, an arm $i_t \in S_1$ (which corresponds a price $p_t < \frac{1}{2}$) is selected in time step $t \in I_2$. Since in time step $t \in I_2$ we always have $v_t > \frac{3}{4}$, thus the revenue loss is at least $\frac{1}{4}$ for each time step $t \in I_2$ with constant probability, thus $\Omega(m)$ in expectation in total. Similarly, if $W_{2mk+1,1} < W_{2mk+1,2}$ at the beginning of I, then at each time step $t \in I_1$ we have $\frac{1}{4} < v_t \leq \frac{1}{2}$, but with constant probability an arm in S_2 is selected and results in $p_t > \frac{1}{2}$ and revenue loss $v_t > \frac{1}{4}$, thus $\Omega(m)$ in expectation in total. To summarize, in either $I_1 = [2mk + \frac{m}{4} + 1, 2mk + \frac{m}{2}]$ or $I_2 = [2mk + \frac{3m}{4} + 1, 2mk + m]$, the total revenue loss is $\Omega(m)$ in expectation. Summing up the revenue loss for all phases, the total revenue loss for Algorithm 12 is $\Omega(T)$ for all time steps, thus $\Omega(1)$ on average.

$\square$

B Buyer's Changing Rate is Dynamic, Known

In this section, we study the problem where ϵ_t are publicly known by the seller. In this case, the algorithms are much simpler than the case in the main body where ϵ_t are unknown to the seller, and we are able to handle the case where ϵ_t are not decreasing for the revenue loss.

Theorem B.1. *If the buyer has adversarial values and changing rates ϵ_t known to the seller, then there exists an online pricing algorithm with symmetric loss $O(\bar{\epsilon}) = O(\frac{1}{T}\sum_{t=1}^{T}\epsilon_t)$. Furthermore, no online algorithm can obtain a symmetric loss less than $\Omega(\bar{\epsilon})$.*

Proof. The tightness result $\Omega(\bar{\epsilon})$ was shown in Theorem 4.1 even when ϵ_t are identical. Now we show that a binary search algorithm that tracks the confidence bound gets symmetric loss $O(\bar{\epsilon})$. We only slightly modify Algorithm 1 as follows. The algorithm maintains confidence bound $[\ell_t, r_t]$ at each time step, and price at $p_t = \frac{\ell_t + r_t}{2}$. If $v_t < p_t$, then $[\ell_{t+1}, r_{t+1}] = [\ell_t - \epsilon_t, p_t + \epsilon_t]$; if $v_t \geq p_t$, then $[\ell_{t+1}, r_{t+1}] = [p_t - \epsilon_t, r_t + \epsilon_t]$.

Let $a_t = r_t - \ell_t$ be the length of the confidence interval at each time. The symmetric loss at time t is upper bounded by $\frac{a_t}{2}$. Observe that $a_{t+1} \leq \frac{a_t}{2} + 2\epsilon_t$, we have

$$a_{t+1} \leq 2\epsilon_t + \epsilon_{t-1} + 2^{-1}\epsilon_{t-2} + \cdots + 2^{-(t-1)}\epsilon_1 + 2^{-(t-1)}a_1. \tag{4}$$

Notice that the total symmetric loss is upper bounded by $\sum_{t=1}^{T} \frac{1}{2}a_t$. When we apply (4) to the formula, the coefficient of any ϵ_t is at most $2 + 1 + 2^{-1} + \cdots \leq 4$. Thus the symmetric loss is $O(\bar{\epsilon})$. □

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

Proof. The tightness result has been shown in Theorem 4.2. Now we construct an online pricing algorithm with revenue loss $\tilde{O}(\bar{\epsilon}^{1/2})$. The algorithm is obtained by slightly modifying Algorithm 3. Firstly, the update rule of confidence interval is changed to $[\ell_{t+1}, r_{t+1}] \leftarrow [\ell_t - \epsilon_t, r_t + \epsilon_t]$. Secondly, the length of each phase is no longer a fixed length. A phase starting at time step $t_0 + 1$ has length m such that $\sum_{t=t_0+1}^{t=t_0+m-1} \epsilon_t \leq \sqrt{\bar{\epsilon}} < \sum_{t=t_0+1}^{t=t_0+m} \epsilon_t$. In other words, the total possible value change in each phase is $\Theta(\sqrt{\bar{\epsilon}})$, so the revenue loss of each step is $O(\sqrt{\bar{\epsilon}})$. Thirdly, in the initialization step of each phase, the algorithm uses binary search algorithm to narrow down the length of the confidence interval to $O(\bar{\epsilon}^{1/2})$.

The number of phases is $O(T\bar{\epsilon}^{-1/2})$ since $\sum_t \epsilon_t = T\bar{\epsilon}$. Then the revenue loss of all phases is contributed by the loss of binary search for locating the value to a confidence interval of length $\sqrt{\bar{\epsilon}}$ in each phase, which is $O(\log \frac{1}{\sqrt{\bar{\epsilon}}}) = \tilde{O}(1)$ per phase and $\tilde{O}(T\bar{\epsilon}^{-1/2})$ in total; plus the revenue loss of each step in the phase, which is at most $O(\sqrt{\bar{\epsilon}})$ each step thus $O(T\sqrt{\bar{\epsilon}})$ in total. Thus the total revenue loss of the algorithm is $\tilde{O}(T\bar{\epsilon}^{-1/2})$ for all time steps, thus $\tilde{O}(\bar{\epsilon}^{-1/2})$ on average. □

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

Proof. The tightness result has been shown in Theorem 4.3. Now we construct an online pricing algorithm with revenue loss $\tilde{O}(\tilde{\epsilon}^{2/3})$. The algorithm is obtained by slightly modifying Algorithm 4. Firstly, the update rule of confidence interval is changed to $[\ell_{t+1}, r_{t+1}] \leftarrow [\ell_t - \epsilon_t, r_t + \epsilon_t]$. Secondly, the length of each phase is no longer a fixed length. A phase starting at time step $t_0 + 1$ has length m such that $\sum_{t=t_0+1}^{t=t_0+m-1} \epsilon_t^2 \leq \tilde{\epsilon}^{4/3} < \sum_{t=t_0+1}^{t=t_0+m} \epsilon_t^2$. Thirdly, let $\delta = 4\tilde{O}(\tilde{\epsilon}^{-2/3})$, then the fixed price in the phase is $p_t = \ell_{t_0} - \delta$. Finally, in the initialization step of each phase, the algorithm uses binary search algorithm to narrow down the length of the confidence interval to $O(4\tilde{\epsilon}^{2/3})$. This is achievable since for any t in the phase $\epsilon_t \leq \tilde{\epsilon}^{2/3}$, as $\sum_{t=t_0+1}^{t=t_0+m-1} \epsilon_t^2 \leq \tilde{\epsilon}^{4/3}$.

The parameters are selected to match the analysis of Algorithm 4. Since $\sum_t \epsilon_t^2 = T\tilde{\epsilon}^2$, there are $\frac{T\tilde{\epsilon}^2}{\Omega(\tilde{\epsilon}^{4/3})} = O(T\tilde{\epsilon}^{2/3})$ phases. In each phase, the binary search incurs $\tilde{O}(1)$ loss, thus $\tilde{O}(T\tilde{\epsilon}^{2/3})$ in all phases. Suppose that at time t_1 the search algorithm ends, and we get an estimate of v_{t_1} with additive accuracy $4\tilde{\epsilon}^{2/3}$. In the remaining of the phase, the price is fixed to be $\ell - \delta$. For any $t \in [t_1, t_0 + m]$, by Azuma's inequality, the probability that $v_t \notin [v_{t_1} - \delta, v_{t_1} + \delta]$ is

$$
\begin{aligned}
\Pr[v_t \notin [v_{t_1} - \delta, v_{t_1} + \delta]] &\leq 2\exp\left(-\frac{\delta^2}{2\sum_t \epsilon_t^2}\right) \\
&= 2\exp\left(-\frac{\delta^2}{2\tilde{\epsilon}^{4/3}}\right) < \frac{1}{m^2}.
\end{aligned}
$$

By union bound, the probability that exists $t \in [t_1, t_0 + m]$ such that $v_t \notin [v_{t_1} - \delta, v_{t_1} + \delta]$ is at most $\frac{1}{m}$; in this case, the revenue loss in the phase is at most 1 per step thus m in total, so the expected revenue loss contributed from the case is $O(1)$. Since there are $O(T\tilde{\epsilon}^{2/3})$ phases, the revenue loss of this case in all phases is $\tilde{O}(T\tilde{\epsilon}^{2/3})$ in total. If $v_t \in [v_{t_1} - \delta, v_{t_1} + \delta]$ for every $t \in [t_1, t_0 + m]$, the revenue loss for setting price $\ell - \delta \geq v_t - 4\tilde{\epsilon}^{2/3} - \delta$ is at most $O(\tilde{\epsilon}^{2/3} + \delta) = O(\delta)$ for each step, thus $O(T\delta)$ in total for all phases with $\ell - \delta \geq v_t - 4\tilde{\epsilon}^{2/3} - \delta$ in each time step. Combine all cases above, the expected revenue loss is $\tilde{O}(T\tilde{\epsilon}^{2/3})$ in total, thus $\tilde{O}(\tilde{\epsilon}^{2/3})$ on average. $\qquad\square$