Proceedings of Machine Learning Research vol 65:1–27, 2017

Bandits with Movement Costs and Adaptive Pricing

Tomer Koren TKOREN@GOOGLE.COM
Google

Roi Livni RLIVNI@CS.PRINCETON.EDU
Princeton University, Computer Science Department

Yishay Mansour MANSOUR@TAU.AC.IL
Tel Aviv University, Blavatnik School of Computer Science

Abstract

We extend the model of Multi-Armed Bandit with unit switching cost to incorporate a metric between the actions. We consider the case where the metric over the actions can be modeled by a complete binary tree, and the distance between two leaves is the size of the subtree of their least common ancestor, which abstracts the case that the actions are points on the continuous interval $[0, 1]$ and the switching cost is their distance. In this setting, we give a new algorithm that establishes a regret of $\widetilde{O}(\sqrt{kT} + T/k)$, where k is the number of actions and T is the time horizon. When the set of actions corresponds to whole $[0, 1]$ interval we can exploit our method for the task of bandit learning with Lipschitz loss functions, where our algorithm achieves an optimal regret rate of $\widetilde{\Theta}(T^{2/3})$, which is the same rate one obtains when there is no penalty for movements.

As our main application, we use our new algorithm to solve an adaptive pricing problem. Specifically, we consider the case of a single seller faced with a stream of patient buyers. Each buyer has a private value and a window of time in which they are interested in buying, and they buy at the lowest price in the window, if it is below their value. We show that with an appropriate discretization of the prices, the seller can achieve a regret of $\widetilde{O}(T^{2/3})$ compared to the best fixed price in hindsight, which outperform the previous regret bound of $\widetilde{O}(T^{3/4})$ for the problem.

1. Introduction

Multi-Armed Bandit (MAB) is a well studied model in computational learning theory and operations research. In MAB a learner repeatedly selects actions and observes their rewards. The goal of the learner is to minimize the regret, which is the difference between her loss and the loss of the best action in hindsight. This simple model already abstracts beautifully the *exploration-exploitation* tradeoff, and allows for a systematic study of this important issue in decision making. The basic results for MAB show that even when an adversary selects the sequence of losses, the learner can guarantee a regret of $\Theta(\sqrt{kT})$, where k is the number of actions and T is the number of time steps (Auer et al., 2002; Audibert and Bubeck, 2009; see also Bubeck and Cesa-Bianchi, 2012).

The simplicity of the MAB comes at a price. Essentially, the system is stateless, and previous actions have no influence on the losses assigned to actions in the future. A more involved model of sequential decision making is Markov Decision Processes (MDPs) where the environment is modeled by a finite set of states, and actions are not only associated with losses but also with stochastic transitions between states. Unfortunately, for the adversarial setting there are mostly hardness results even in limited cases (Abbasi et al., 2013).

Introducing switching costs is a step of incorporating dependencies in the learner's action selection. The unit switching cost has a unit cost per each changing of actions. In such a setting a tight bound of $\widetilde{\Theta}(k^{1/3}T^{2/3})$ is known (Dekel et al., 2014a). Our main goal is to extend this basic model to the case of MAB with movement costs, where the cost associated with switching between arms is given by a metric that determines the distance between any pair of arms. Such a model already introduces a very interesting dependency in the action selection process for the learner. Specifically, we study a metric between actions which is modeled by a complete binary tree, where the distance between two actions is proportional to the number of nodes in the subtree of their least common ancestor. This abstracts the case where the arms are associated with k points on the real line and the switching cost between arms is the absolute difference between the corresponding points (actually, the tree metric only upper bounds distances on a line, but this upper bound is sufficient for our applications). Note that we do not assume that pairs of actions with low movement cost have similar losses: our model retains the full generality of the loss functions, and only imposes a metric structure on the cost of movement between arms.

Our main result is an efficient MAB algorithm, called the Slowly Moving Bandit (SMB) algorithm, that guarantees expected regret of at most $\widetilde{O}(\sqrt{kT} + T/k)$. As we elaborate later, this result implies that for $k \leqslant T^{1/3}$ we can achieve an optimal regret $\widetilde{\Theta}(T^{2/3})$, and for $k \geqslant T^{1/3}$ we obtain an optimal regret rate of $\widetilde{\Theta}(\sqrt{kT})$. It is worth discussing the implication of our bound. The bound of $\widetilde{\Theta}(T^{2/3})$ for $k \leqslant T^{1/3}$ is tight due to the lower bound of Dekel et al. (2014a), which applies already for $k = 2$ actions. The bound of $\widetilde{\Theta}(\sqrt{kT})$ for $k \geqslant T^{1/3}$ is tight due to the classic lower bound for MAB even without movement costs (Auer et al., 2002). Surprisingly, for a large action set (i.e., $k \geqslant T^{1/3}$) we lose nothing in the regret by introducing movement costs to the problem! Another surprising consequence of our bound is that there is no loss in the regret by increasing the number of actions from $k = 2$ to $k = \Theta(T^{1/3})$ when movement costs are present.

The main application of our SMB algorithm is for adaptive pricing with patient buyers (Feldman et al., 2016). In this adaptive pricing problem, we have a seller who would like to maximize his revenue. He is faced with a stream of patient buyers. Each buyer has a private value and a window of time in which she would like to purchase the item. The buyer buys at the lowest price in its window, in case this price is below the buyer's private valuation of the item. (The seller publishes prices sufficient far into the future, so that the buyer can observe all the relevant prices.) The adaptive price setting is related to the MAB problem with movement costs in the following way. The prices are continuous (say, $[0, 1]$) and the reward is the revenue gained by the seller. The rewards are given by a one–sided Lipschitz function (specifically, we receive the reward whenever we post a price which is at most the private value, and zero otherwise). This allows us to apply our bandit algorithm via discretization of the continuous space. The challenge, though, remains to control the cost the seller pays which stems from the buyer's patience.

The seller benchmark is the best single price. Using a single price implies that the buyers either buy immediately, or never buy. The movement cost models the loss due to having the buyer patient, which can be thought as the difference between the price of the item when the buyer arrives and the price at which it buys. (Note that there might be a gain, since it might be that when the buyer arrives the price is too high, but later lower prices make him buy. We ignore this effect for now.) Our main result is that the seller can use our SMB algorithm and guarantee a regret of at most $\widetilde{O}(T^{2/3})$, using $T^{1/3}$ equally-spaced prices. This is in contrast to a regret of $\widetilde{O}(T^{3/4})$ which is achieved by applying a standard switching cost technique together with a discretization argument (Feldman et al., 2016).

It is interesting to observe qualitatively how our algorithm performs. It is much more likely to make small changes than large ones; roughly speaking, the probability of a change drops exponentially in the magnitude of the change. Conceptually, this is a highly desirable property of a pricing algorithm, and arguably, of any regret minimization algorithm: we would like to slightly perturb the prices over time without a sever impact on the buyers, and only rarely make very large changes in the pricing.

Finally, another application of our algorithm is for the case that we have continuous actions on an interval, and the losses of the actions are Lipschitz. Our algorithm can handle movement cost which are also Lipschitz on the interval. (We stress that in our application the losses are deterministic and not stochastic.)

1.1. Related Work

With a uniform unit switching cost (i.e., when switching between any two actions has a unit cost), it is known that there is a tight $\widetilde{\Omega}(k^{1/3}T^{2/3})$ lower bound for the MAB problem (Dekel et al., 2014a), which is in contrast to the $O(\sqrt{kT})$ regret upper bound without switching costs.

Classical MAB algorithms such as Exp3 (Auer et al., 2002) guarantee a regret of $\widetilde{O}(\sqrt{kT})$ without movement costs. However, they are not guaranteed to move slowly between actions, and in fact, it is known that Exp3 might make $\widetilde{\Omega}(T)$ switches between actions in the worst case (see Dekel et al., 2014a), which makes it inappropriate to directly handle movement costs.

Our adaptive pricing application follows the model of Feldman et al. (2016). There, for a finite set of k prices a matching bound of $\widetilde{\Omega}(T^{2/3})$ on the regret is shown. For continuous prices they remark that their upper bound can be used to derive an $\widetilde{O}(T^{3/4})$ regret bound. Our SMB algorithm improves this regret bound to $\widetilde{O}(T^{2/3})$. There is a slight difference in the exact feedback model between Feldman et al. (2016) and here: in both models when a buyer arrives, the sell time is uniquely determined; however, in Feldman et al. (2016) the seller observes the purchase only at the actual time of the sell, whereas here we assume the seller observes the sell when the buyer arrives and decides when to purchase. We remark, though, that as discussed in Feldman et al. (2016) all lower bounds derived there apply to the current feedback model too.

There is a vast literature on online pricing (e.g., Balcan and Blum, 2006; Balcan et al., 2008; Balcan and Constantin, 2010; Bansal et al., 2010; Besbes and Zeevi, 2009). The main difference of our adaptive pricing model is the patience of our buyers, which correlates between the prices at nearby time steps.

For the case of continuous prices and a single seller, when one consider *impatient* buyers, a simple discretization argument can be used to achieve a regret of $\widetilde{O}(T^{2/3})$, and there exists a similar lower bound of $\Omega(T^{2/3})$ (Kleinberg and Leighton, 2003). More generally, learning

Lipschitz functions on a closed interval has been studied by Kleinberg (2004), where an optimal $\widetilde{\Theta}(T^{2/3})$ regret bound is shown via discretization. Our results show that even if one adds a movement cost (which is the distance) to the problem, there is no change in the regret.

There are many works on continuous action MAB (Kleinberg, 2004; Cope, 2009; Auer et al., 2007; Bubeck et al., 2011; Yu and Mannor, 2011). Most of the works relate the change in the payoff to the change in the action in various ways. Specifically, there is an extensive literature on the Lipschitz MAB problem and various variants thereof (Kleinberg et al., 2008; Slivkins, 2011; Slivkins et al., 2013; Kleinberg and Slivkins, 2010), where the expectation of the reward of arms have a Lipschitz property. We differ from that line of work. Our assumption is about the switching cost (rather than the losses) being related to the distance between the actions.

The work of Guha and Munagala (2009) discusses a stochastic MAB, in the spirit of the Gittins index, where there is both a switching cost and a play cost, and gives a constant approximation algorithm. We differ from that work both in the model, their model is stochastic and our is adversarial, and in the result, their is a multiplicative approximation and our is a regret.

Approximating an arbitrary metric using randomized trees (i.e., k-HST) has a long history in the online algorithms literature, starting with the work of Bartal (1996). The main goal is to derive a simpler metric representation (using randomized trees) that will both upper and lower bound the given metric. In this work we need only an upper bound on the metric, and therefore we can use a deterministic complete binary tree.

2. Setup and Formal Statement of Results

2.1. Bandits with Movement Costs

In this section we consider the Multi-Armed Bandit (MAB) problem with movement costs. In this problem, that can be described as a game between an online learner and an adversary continuing for T rounds, where there is a set $K = \{1, \ldots, k\}$ of $k \geqslant 2$ arms (or actions) that the learner can choose from. The set of arms is equipped with a metric $\Delta(i, j) \in [0, 1]$ that determines the movement distance between any pair of arms $i, j \in K$.

First, before the game begins, the adversary fixes a sequence $\ell_1, \ldots, \ell_T \in [0, 1]^k$ of loss vectors assigning loss values in $[0, 1]$ to the arms.[1] Then, on each round $t = 1, \ldots, T$, the learner picks an arm $i_t \in K$, possibly at random, and suffers the associated loss $\ell_t(i_t)$. In addition to incurring this loss, the learner also pays a cost of $\Delta(i_t, i_{t-1})$ that results from her movement from arm i_{t-1} to arm i_t. At the end of each round t, the learner receives *bandit feedback*: she gets to observe the single number $\ell_t(i_t)$, and this number only. (The movement cost is common knowledge.)

The goal of the learner, over the course of T rounds of the game, is to minimize her expected movement-regret, which is defined as the difference between her (expected) total costs—including both the losses she has incurred as well as her movement costs—and the total costs of the best fixed action in hindsight (that incur no movement costs, since it is

1. Throughout, we assume that the adversary is *oblivious*, namely, that it cannot react to the learner's actions.

the same action in all time steps); namely, the *movement regret* with respect to a sequence $\ell_{1:T}$ of loss vectors and the metric Δ equals

$$\mathrm{Regret}_{\mathsf{MC}}(\ell_{1:T}, \Delta) = \mathbb{E}\left[\sum_{t=1}^{T} \ell_t(i_t) + \sum_{t=2}^{T} \Delta(i_t, i_{t-1})\right] - \min_{i^\star \in K} \sum_{t=1}^{T} \ell_t(i^\star) \ .$$

Here, the expectation is taken with respect to the player's randomization in choosing the actions $i_1, \ldots, i_T$.

MAB with a tree metric. Our focus in this paper is on a metric induced over the actions by a complete binary tree $\mathcal{T}$ with k leaves. We consider the MAB setting where each action i is associated with a leaf of the tree $\mathcal{T}$. (For simplicity, we assume that k is a power of two.)

We number the levels of the tree $\mathcal{T}$ from the leaves to the root. Let $\mathsf{level}(v)$ be the level of node v in $\mathcal{T}$, where the level of the leaves is 0. Given two leaves i and j, let $\mathsf{lca}(i,j)$ be their least common ancestor in $\mathcal{T}$. Then, given actions i and j let $d_{\mathcal{T}}(i,j)$ be the level of their least common ancestor in $\mathcal{T}$, i.e., $d_{\mathcal{T}}(i,j) = \mathsf{level}(\mathsf{lca}(i,j))$. The movement cost between i and j is then

$$\Delta_{\mathcal{T}}(i,j) = \tfrac{1}{k} 2^{d_{\mathcal{T}}(i,j)} \in [0,1] \ . \tag{1}$$

Our first main result bounds that movement cost with respect to the given metric:

Theorem 1. *There exists an algorithm (see Algorithm 1 in Section 4) that for any sequence of loss functions $\ell_1, \ldots, \ell_T$ guarantees that*

$$\mathit{Regret}_{\mathsf{MC}}(\ell_{1:T}, \Delta_{\mathcal{T}}) = \widetilde{O}\left(\sqrt{kT} + \frac{T}{k}\right) \ .$$

For $k \geqslant T^{1/3}$ the theorem gives an optimal regret bound of $\widetilde{O}(\sqrt{kT})$. For $k \leqslant T^{1/3}$, we can extend a binary tree with k leaves by turning each leaf into a node whose subtree is a balanced binary tree and we obtain a new tree with at most $2T^{1/3}$ leaves. We then associate with each new leaf as its action the action induced by its parent at the level of original leaves. One can show that the movements between the level of the original actions is then controlled by $O(T^{2/3})$ and we can then exploit this construction to achieve a regret bound of $\widetilde{O}(T^{2/3})$. In any movement cost problem with at least two arms of fixed constant distance, a lower bound regret of 2-arm switching cost applies, hence we observe that these rates are optimal for every $k \leqslant T$ (Dekel et al., 2014a).

The proof of Theorem 1 is provided in Section 4 as a direct corollary of Theorem 4.

Continuum-armed bandit with movement cost. We can apply Algorithm 1 to the problem of learning Lipschitz functions over the real line with movement regret associated with standard metric over the interval. In this setting we assume an arbitrary sequence of functions $f_1, \ldots, f_T : [0,1] \mapsto [0,1]$ where each function f_t is L-Lipschitz. i.e.,

$$|f_t(x) - f_t(y)| \leqslant L|x-y| \qquad\qquad \forall\ x, y \in [0,1] \ .$$

Let x_t be the action selected by the player at time t. The objective is then to minimize the *movement regret*, defined:

$$\text{Regret}_{\mathsf{MC}}(f_{1:T}, |\cdot|) = \mathbb{E}\left[\sum_{t=1}^{T} f_t(x_t) + \sum_{t=1}^{T} |x_t - x_{t+1}|\right] - \min_{x \in [0,1]} \sum_{t=1}^{T} f_t(x) \ .$$

One application of our algorithm is a regret bound for Lipschitz functions:

Theorem 2. *There exists an algorithm (based on Algorithm 1) that for every sequence of L-Lipschitz loss functions $f_1, \ldots, f_T$, with $L \geqslant 1$, achieves:*

$$\mathit{Regret}_{\mathsf{MC}}(f_{1:T}, |\cdot|) = \widetilde{O}\big(L^{1/3} T^{2/3}\big) \ .$$

We emphasize that even without movement costs, there is an $\widetilde{\Omega}(T^{2/3})$ lower bound in this setting (Kleinberg, 2004); hence, the regret bound of Theorem 2 is essentially optimal.

Section 4.6 is dedicated to the proof of Theorem 2. We also note that the result, in fact, holds for any metric Δ that is L-Lipschitz (for exact statement see Theorem 12).

2.2. Adaptive Pricing

We consider the following model of online learning, with respect to a stream of patient buyers with patience at most $\overline{\tau}$. In our setting the seller posts at time $t = 1$ prices $\rho_1, \ldots, \rho_{\overline{\tau}+1}$ for the next $\overline{\tau}$ days in advance. Then at each time step t the seller posts price for the $t + \overline{\tau}$ day $\rho_{t+\overline{\tau}}$ and receives as feedback her revenue for day t. The revenue at time t depends on buyer $\mathbf{b}_t$ and the sequence of prices $\rho_t, \rho_{t+1}, \ldots, \rho_{t+\overline{\tau}}$ in the manner described below.

Each buyer $\mathbf{b}_t$, in our setting, is a mapping from a sequence of prices to revenues, parameterized by her *value* v_t and her patience τ_t. The buyer proceed by observing prices $\rho_t, \ldots, \rho_{t+\tau_t}$, and purchases the item at the lowest price among these prices, if it does not exceed her value. Thus the revenue from the buyer at time t is described as follows:

$$\mathbf{b}_t(\rho_t, \ldots, \rho_{t+\overline{\tau}}) = \begin{cases} \min\{\rho_t, \ldots, \rho_{t+\tau_t}\} & \text{if } \min\{\rho_t, \ldots, \rho_{t+\tau_t}\} \leqslant v_t, \\ 0 & \text{otherwise.} \end{cases}$$

Note that at time t the buyer decides whether it will purchase and when. Here, we assume that the buyer also gets to order the good at day of arrival (at price and time decided by him according to his patience and private value), thus the seller observes the buyer's decision at time t, namely the feedback at time t is given by $\mathbf{b}_t(\rho_t, \ldots, \rho_{t+\overline{\tau}})$. We note that this feedback model differs from Feldman et al. (2016) where the buyer buy at day of purchase. However, we note that both lower and upper bounds derived by Feldman et al. apply to our feedback model as noted there in the discussion.

Our objective is to construct an algorithm that minimizes the regret which is the difference between revenue obtained by the best fixed price in hindsight and the expected revenue obtained by the seller, given a sequence $\mathbf{b}_{1:T}$ of buyers:

$$\text{Regret}(\mathbf{b}_{1:T}) = \max_{\rho^* \in P} \sum_{t=1}^{T} \mathbf{b}_t(\rho^*, \ldots, \rho^*) - \mathbb{E}\left[\sum_{t=1}^{T} \mathbf{b}_t(\rho_t, \ldots \rho_{t+\overline{\tau}})\right].$$

Our main result with respect to adaptive pricing is as follows:

Theorem 3. *There exists an algorithm (see Algorithm 2 in Section 5) that for any sequence of buyers $\mathbf{b}_1, \ldots, \mathbf{b}_T$ with maximum patience $\overline{\tau}$ achieves the following regret bound:*

$$\textit{Regret}(\mathbf{b}_{1:T}) = \widetilde{O}(\overline{\tau}^{1/3} T^{2/3}) .$$

It is interesting to note that even though a lower bound of $\Omega(T^{2/3})$ stems from two different sources we can still achieve a regret rate of $\widetilde{O}(T^{2/3})$. Indeed, Kleinberg and Leighton (2003) showed that optimizing over the continuum $[0, 1]$ leads to a lower bound of $\Omega(T^{2/3})$, irrespective of the patience of the buyers. Second, Feldman et al. (2016) showed that whenever the seller wishes to optimize between more than two prices, a lower bound of $\Omega(T^{2/3})$ holds for patient buyers.

In this work we deal with both obstacles together—patient buyers and optimization over the $[0, 1]$ interval—yet the two obstacles can be dealt without leading to a regret bound that is necessarily worse then each obstacle alone.

Our solution to the adaptive pricing problem is based on employing a MAB with movement costs algorithm that allows small change in the prices. The reason one needs to employ an algorithm with small movement cost stems from the memory of the buyers: roughly speaking, whenever the seller encounters a buyer with patience, the potential revenue of the seller will be the revenue at time t minus any discount price that buyer may encounter on future days. Indeed, for the case of two prices, Feldman et al. (2016) constructed a sequence of buyers that reduces the problem to MAB with switching cost: a step in demonstrating a $\Omega(T^{2/3})$ regret bound: thus a fluctuation in prices is indeed a cause for a high regret.

A full proof of Theorem 3 is provided in Section 5.

3. Overview of the approach and techniques

In this section we give an informal overview of the main ideas in the paper and describe the techniques used in our solution. We begin with the main ideas behind our main result: an optimal and efficient algorithm for MAB problems with movement costs. Later we continue with the adaptive pricing problem, and show how it is abstracted as an instance of the MAB problem with movement costs.

From continuum-armed to multi-armed. In our main applications, we consider actions that are associated to points on the interval $[0, 1]$ equipped with the natural metric $\Delta(x, y) = |x - y|$. As a preliminary step, we use discretization in order to make the action space finite and capture the setting by the MAB framework. That is, we reduce the problem of minimizing regret over the entire $[0, 1]$ interval to regret minimization over k actions associated with the equally-spaced points $K = \{\frac{1}{k}, \frac{2}{k}, \ldots, 1\}$. Our challenge is to then to design a regret minimization algorithm over $\mathcal{A}$ whose cumulative movement cost with respect to the metric $\widetilde{\Delta}(i, j) = |i - j|/k$ is bounded.

Our approach builds upon the basic techniques underlying the Exp3 algorithm for the basic MAB problem, which we recall here. Exp3 maintains over rounds a distribution p_t over the k actions and chooses an action $i_t \sim p_t$; thereafter, it updates its sampling distribution multiplicatively via $p_{t+1}(i) \propto p_t(i) \cdot \exp(-\eta \overline{\ell}_t(i))$, where $\overline{\ell}_t$ is an unbiased estimator of true loss vector ℓ_t constructed using only the observed feedback $\ell_t(i_t)$. Specifically, the estimator used by Exp3 is

$$\bar{\ell}_t(i) = \frac{\mathbb{1}\{i_t = i\}}{p_t(i)} \ell_t(i_t) \qquad \forall\, i \in K\ .$$

A simple computation shows that $\bar{\ell}_t$ is indeed an unbiased estimator of ℓ_t, namely that $\mathbb{E}[\bar{\ell}_t] = \ell_t$, and the crucial bound for Exp3 is then obtained by controlling a variance term of the form $\mathbb{E}[p_t \cdot \bar{\ell}_t^2]$, and showing that it is of the order $\widetilde{O}(k)$ at all rounds t. This in turn implies the $\widetilde{O}(\sqrt{kT})$ bound of Exp3.

Controlling movements with a tree. As a first step in controlling the movement costs of our algorithm, one can think of an easier problem of controlling the number of times the algorithm switches between actions in the left part of the interval, namely in $A_L = \{\frac{1}{k}, \ldots, \frac{1}{2}\}$, and actions in the right part of the interval, $A_R = \{\frac{1}{2} + \frac{1}{k}, \ldots, 1\}$. Indeed, since each such switch might incur a high movement cost (potentially close to 1), any algorithm for MAB with movement costs must avoid making such switches too often. In principle, a solution to this simpler problem can be then lifted to a solution to the actual movement costs problem by applying it recursively to each side of the interval.

The thought experiment above motivates our tree-based metric: this metric assigns a fixed cost of 1 to any movement between the left and right parts of the interval—that correspond to the topmost left and right subtrees—and recursively, a cost of $2^d/k$ for any movement between subtrees in level d of the tree. The tree metric is always an upper bound on the natural metric on the interval, namely $\widetilde{\Delta}(i,j) \leqslant \frac{1}{k} 2^{d_{\mathcal{T}}(i,j)} = \widetilde{\Delta}_{\mathcal{T}}(i,j)$, so that controlling movement costs with respect to $\widetilde{\Delta}_{\mathcal{T}}$ suffices for controlling movement costs with respect to the natural distance on $[0,1]$. While this upper bound might occasionally be very loose,[2] the tree-metric effectively captures the difficulties of the original movement costs problem with the natural metric over $[0,1]$.

Hence, we henceforth focus on constructing an algorithm with low movement costs with respect to a tree-based metric over a full binary tree. To accomplish this, we will regulate the probability of switching the ancestral node. Namely, if we denote by $A_d(i)$ the subtree at level d of the tree containing action i, our goal is to design an algorithm that switches between actions i and j such that $A_d(i) \neq A_d(j)$ with probability at most 2^{-d}. This would ensure that the expected contribution of level d in the tree to the movement cost of the algorithm is $O(1/k)$ per round. Indeed, switching between subtrees at level d (while not making a switch at higher levels) results with a movement cost of roughly $2^d/k$. Overall, the contribution of all layers in the tree to the total movement cost would then be $O((T/k)\log k)$, as required.

Lazy sampling. Our challenge now is to construct an algorithm that switches infrequently between subtrees at higher levels of the tree. However, recall that typical bandit algorithms choose their actions $i_1, \ldots, i_T$ at random from sampling distributions $p_1, \ldots, p_T$ maintained throughout the evolution of game. In order to guarantee that consecutive actions i_t and i_{t-1} will belong to the same subtree with high probability, the algorithm would have to sample i_i in a way which is highly correlated with the preceding action i_{t-1}.

2. For example, the distance between $\frac{1}{2} - \frac{1}{k}$ and $\frac{1}{2} + \frac{1}{k}$ according to the metric $\Delta_{\mathcal{T}}$ is 1.

Suppose that the marginals of the subtrees at some level d do not change between the distributions p_{t-1} and p_t; namely, that the cumulative probability assigned to the leaves of each such subtree by both p_{t-1} and p_t is the same. In this case, we argue that we can sample our new action i_t at time t, based on the preceding action i_{t-1}, from the conditional distribution $p_t(\cdot \mid A_d(i_{t-1}))$. In other words, if we think of sampling an action i from p_t as sampling a path in the tree leading to the leaf associated with i, then for determining i_t on round t we copy the top d edges from the path at time $t-1$, and only sample the remaining bottom edges (those contained in the subtree $A_d(i_{t-1})$) according to the new distribution p_t. Intuitively, this can be justified because the distribution of the top d edges in the path leading to i_t is the same as that of the top d edges in the path leading to i_{t-1}, so we may as well keep the random bits associated with them and only resample bits associated with the remaining edges from fresh.

The lazy sampling scheme sketched above raises a major difficulty in the analysis: since i_t is sampled from a conditional of p_t that might be very different from p_t itself, it is no longer clear that i_t is distributed according to the "correct" distribution. In other words, conditioned on p_t (which intuitively is a summary of the past), the random variable i_t is certainly not distributed according to p_t. Nevertheless, our analysis demonstrates a crucial property of the distributions p_t maintained the sampling scheme, which is sufficient for the regret analysis: we show that for all subtrees A at all levels of the tree, it holds that

$$\mathbb{E}\left[\frac{\mathbb{1}\{i \in A\}}{p_t(A)}\right] = 1 \ .$$

That is, even though i_t is sampled indirectly from p_t, it is still distributed according to p_t in a certain sense.

Rebalancing the marginals. The lazy sampling we described above reduced the problem of controlling the frequency of movements in the actions $i_1, \ldots, i_T$, to controlling the frequency in which the marginal distribution of $p_1, \ldots, p_T$ over subtrees is updated by our algorithm. Next, we describe how the latter is accomplished (where the frequency of update is exponentially-decreasing with the level of the subtree). To illustrate the technique, let us consider an easier problem: instead of demanding infrequent updates for subtrees in all levels, we shall only attempt to rebalance the marginals at the topmost level, with the goal of making them being updated with probability at most $2^{-D} = 1/k$. We will demonstrate how the estimator $\widetilde{\ell}_t$ can be modified in a way that induces such infrequent updates at the top level. Denote the left subtree at the top level by A_L (containing actions $\frac{1}{k}, \ldots, \frac{1}{2}$) and the right topmost subtree by A_R (containing actions $\frac{1}{2} + \frac{1}{k}, \ldots, 1$). First, we choose

$$\sigma_t = \begin{cases} 1 - \frac{1}{\delta} & \text{with probability } \delta; \\ 1 & \text{with probability } 1 - \delta. \end{cases}$$

Then, for $A \in \{A_L, A_R\}$ we set

$$\widetilde{\ell}_t(i) = \bar{\ell}_t(i) - \frac{\sigma_t}{\eta} \log\left(\sum_{j \in A} \frac{p_t(j)}{p_t(A)} e^{-\eta \bar{\ell}_t(j)}\right) \qquad \forall\ i \in A \ .$$

Here, $\bar{\ell}_t$ is the basic Exp3 estimator discussed earlier. In terms of estimation, $\widetilde{\ell}_t$ is still an unbiased estimator of the true vector ℓ_t: since $\mathbb{E}[\sigma_t] = 0$ it follows that $\mathbb{E}[\widetilde{\ell}_t] = \ell_t$. However, the added term has a balancing effect at the top level of the tree: a simple computation reveals that if $\sigma_t = 1$ (which occurs with high probability), the multiplicative update of the algorithm applied on the vector $\widetilde{\ell}_t$ ensures that $p_t(A_L) = p_{t+1}(A_L)$ and $p_t(A_R) = p_{t+1}(A_R)$. In other words, with probability $1 - \delta$, the cumulative (i.e., marginal) probability of both subtrees at the top level is remained fixed between rounds t and $t + 1$.

The balancing effect we achieved comes at a price: for small values of δ the magnitude of $\widetilde{\ell}_t$ becomes large, as it might be the case that $\sigma_t \approx -1/\delta$. Nevertheless, it is not hard to show that the variance term $\mathbb{E}[p_t \cdot \widetilde{\ell}_t^2]$ is bounded by $O(k + 1/\delta)$. In particular, for $\delta = 1/k$ we retain a variance bound of $O(k)$, while changing the marginals of the two top subtrees with probability no larger than $1/k$. As a result, by sampling accordingly from the slowly-changing distributions p_t we can ensure that the movements at the top level contribute at most $O(T/k)$ to the total movement cost of the algorithm.

Evidently, the estimator described above only remedies the problem at the top level, and the movement costs at lower levels of the tree might still be very large (effectively, within each subtree the algorithm does nothing but simulating Exp3 on the leaves). Still, using a similar yet more involved technique we can induce a balancing effect at all levels simultaneously and ensure that the marginal probabilities of the subtrees at level d are modified by the algorithm with probability at most 2^{-d}. The construction adds a balancing term corresponding to each level of the tree in a recursive manner that takes into account the balancing terms at lower levels.

From adaptive pricing to bandits. We now discuss how to reduce adaptive pricing with patient buyers to a MAB problem with movement costs. We employ a reduction similar to the one used by Kleinberg and Leighton (2003); however, the patience of the buyers introduce some difficulties, as we discuss below. For now, we ignore the buyers' patience and give the idea of the reduction in the simplest case.

Intuitively, in order to adaptively pick prices from the interval $[0, 1]$ so as to minimize regret with respect to the best fixed price in hindsight, we could directly apply a standard MAB algorithm, e.g., Exp3, over a discretization $\mathcal{A} = \{\frac{1}{k}, \frac{2}{k}, \ldots, 1\}$ of the interval, treating each of the k prices as an arm that generates a reward whenever it is pulled. Furthermore, since the buyers' valuations are not disclosed after purchase, the feedback observed by the seller is very limited and nicely captured by the MAB abstraction. Since the buyers' valuations are one-sided Lipschitz, the best price in $\mathcal{A}$ will lose at most $O(T/k)$ in total revenue as compared to the best fixed price in the entire $[0, 1]$ interval. Thus, provided an algorithm that achieves $\widetilde{O}(\sqrt{kT})$ expected regret with respect to the best price in $\mathcal{A}$, we could pick $k = \Theta(T^{1/3})$ and obtain the optimal $\widetilde{O}(T^{2/3})$ regret for the pricing problem.

Patient buyers and movement costs. A main complication in the above MAB approach arises from the buyers' patience: the revenue extracted from a single buyer is determined not only by the price posted by the seller on the day of the buyer's arrival, but also by prices posted on the subsequent days subject to the buyer's patience. As a result, if the seller change prices abruptly on consecutive days, a strategic buyer—that purchases in the minimal price, if at all—could make use of this fact to gain the item at a lower price, which lowers the revenue of the seller. Roughly speaking, the latter additional cost to the seller is controlled by the absolute difference between the prices she posted at consecutive days. Thus, the pricing problem with patient buyers can be reduced to a MAB problem with movement costs, where the online player suffers an additional movement cost each time she changes actions, and the movement cost is determined by the metric (absolute value distance) between the respective actions.

The reduction sketched above is made precise in Section 5, where we also address an additional difficulty stemming from the adaptivity of the *feedback signal* observed by the seller: the latter is contaminated by the effect of prices posted at earlier rounds on the buyers, and has to be treated carefully.

4. The Slowly Moving Bandit Algorithm

In this section we present the Slowly Moving Bandit (SMB) algorithm: our optimal algorithm for the Multi-armed bandit problem with movement costs.

In order to present the algorithm we require few additional notations. Recall that in our setting, we consider a complete binary tree of depth $D = \log_2 k$ whose leaves are identified with the actions $1, \ldots, k$ (in this order). For any level $0 \leqslant d \leqslant D$ and arm $i \in K$, let $A_d(i)$ be the set of leaves that share a common ancestor with i at level d (where level $d = 0$ are the singletons). We denote by $\mathcal{A}_d$ the collection of all $k/2^d$ subsets of leaves:

$$\mathcal{A}_d = \Big\{\{1, \ldots, 2^d\}, \{2^d + 1, \ldots, 2 \cdot 2^d\}, \ldots, \{k - 2^d + 1, \ldots, k\}\Big\} \qquad \forall\ 0 \leqslant d \leqslant D\ .$$

The SMB algorithm is presented in Algorithm 1. The algorithm is based on the multiplicative update method, and in that sense is reminiscent of the EXP3 algorithm (Auer et al., 2002). Similarly to EXP3, the algorithm computes at each round t an estimator $\widetilde{\ell}_t$ to the true, unrevealed loss vector ℓ_t using the single loss value $\ell_t(i_t)$ observed on that round.

As discussed in Section 3, in addition to being an (almost) unbiased estimate for the true loss vector, the estimator $\widetilde{\ell}_t$ used by SMB has the additional property of inducing slowly-changing sampling distributions p_t, that allow for sampling the actions i_t in a way that the overall movement cost is controlled. This is achieved by choosing at random, at each round t, a level d_t of the tree to be rebalanced by the algorithm using the balancing vectors $\overline{\ell}_{t,d}$. For reasons that will become apparent later on, the level d_t is determined by choosing a random sign $\sigma_{t,d}$ for each level d in the tree and identifying the bottommost level with a negative sign, namely

$$d_t = \min\{0 \leqslant d \leqslant D : \sigma_{t,d} < 0\}, \quad \text{where} \quad \sigma_{t,D} = -1.$$

Then, as we show in the analysis, the terms $\overline{\ell}_{t,d}$ defined using the signs $\sigma_{t,d}$ have a balancing effect at levels $d \geqslant d_t$.

A major difficulty inherent to our approach, also common to many bandit optimization settings (e.g., Dani et al., 2007; Alon et al., 2015; Bubeck et al., 2016), is the fact that the estimated losses $\widetilde{\ell}_t(i)$ might receive negative values that are very high in absolute value. Indeed, the balancing term $\overline{\ell}_{t,d}$ corresponding to level d is roughly as large as $2^d/p_t(i_t)$, and might appear with a negative sign in $\widetilde{\ell}_t$. Algorithm 1 resolves this issue by zeroing-out the estimator $\widetilde{\ell}_t$ whenever it chooses an action whose probability is too small, which ensures that the $\overline{\ell}_{t,d}$ terms never become too large. We remark that the standard approaches used to resolve such issues (the simplest of which is mixing the distribution p_t with the uniform distribution over the k actions) fail in our case, as they break the rebalancing effect which is tailored to the specific multiplicative update of the algorithm.

Initialize $p_1 = u$, $d_0 = D$ and $i_0 \sim p_1$; for $t = 1, \ldots, T$:

(1) Choose action $i_t \sim p_t(\,\cdot \mid A_{d_{t-1}}(i_{t-1}))$, observe loss $\ell_t(i_t)$

(2) Choose $\sigma_{t,0}, \ldots, \sigma_{t,D-1} \in \{-1, +1\}$ uniformly at random

(3) Compute vectors $\overline{\ell}_{t,0}, \ldots, \overline{\ell}_{t,D-1}$ recursively via

$$\overline{\ell}_{t,0}(i) = \frac{\mathbb{1}\{i_t = i\}}{p_t(i)} \ell_t(i_t) \ ,$$

and for all $d \geqslant 1$:

$$\overline{\ell}_{t,d}(i) = -\frac{1}{\eta} \log \left(\sum_{j \in A_d(i)} \frac{p_t(j)}{p_t(A_d(i))} e^{-\eta(1+\sigma_{t,d-1})\overline{\ell}_{t,d-1}(j)} \right)$$

(4) Define $B_t = \{i \in K : p_t(A_d(i)) < 2^d \eta \text{ for some } 0 \leqslant d < D\}$ and set

$$\widetilde{\ell}_t = \begin{cases} 0 & \text{if } i_t \in B_t; \\ \overline{\ell}_{t,0} + \sum_{d=0}^{D-1} \sigma_{t,d} \overline{\ell}_{t,d} & \text{otherwise} \end{cases}$$

(5) Update:

$$p_{t+1}(i) = \frac{p_t(i)\, e^{-\eta \widetilde{\ell}_t(i)}}{\sum_{j=1}^k p_t(j)\, e^{-\eta \widetilde{\ell}_t(j)}} \qquad \forall\ i \in K$$

Algorithm 1: The SMB algorithm.

The following theorem is the main result of this section. Theorem 1 is an immediate corollary.

Theorem 4. *For any sequence of loss functions $\ell_1, \ldots, \ell_T$, The* SMB *algorithm (Algorithm 1) guarantees that*

$$\mathit{Regret}(\ell_{1:t}) = O\left(\frac{\log k}{\eta} + \eta T k \log k \right).$$

In particular, by setting $\eta = 1/\sqrt{kT}$ the expected regret of the algorithm is bounded by $O(\sqrt{Tk} \log k)$. Furthermore, for the metric $\Delta_{\mathcal{T}}$ (see Eq. (1)), the expected total movement cost of the algorithm is $\mathbb{E}[\sum_{t=2}^T \Delta_{\mathcal{T}}(i_t, i_{t-1})] = O((T/k) \log k)$.

The rest of the section focuses on proving Theorem 4. We begin by stating a useful technical bound that we use throughout our analysis to control the magnitude of the balancing vectors $\overline{\ell}_{t,d}$. For a proof of the lemma, see Section 4.5 below.

Lemma 5. *For all t and $0 \leqslant d < D$ the following holds almost surely:*

$$0 \leqslant \overline{\ell}_{t,d}(i) \leqslant \frac{\mathbb{1}\{i_t \in A_d(i)\}}{p_t(A_d(i))} \prod_{h=0}^{d-1}(1+\sigma_{t,h}) \qquad \forall\ i \in K\,. \tag{2}$$

In particular, if $\sigma_{t,h} = -1$ then $\overline{\ell}_{t,d} = 0$ for all $d > h$.

One useful implication of the lemma is that, since $\overline{\ell}_{t,d} = 0$ for all $d > d_t$, we can express our estimator $\widetilde{\ell}_t$ in the following equivalent form:

$$\widetilde{\ell}_t = \overline{\ell}_{t,0} - \overline{\ell}_{t,d_t} + \sum_{h=0}^{d_t-1} \overline{\ell}_{t,h}\ . \tag{3}$$

4.1. Rebalancing the marginals

Our first step is to show that the marginals of the distributions p_t over subtrees of actions are not modified by the algorithm with high probability, as a result of adding the balancing vectors $\overline{\ell}_{t,d}$.

Lemma 6. *For all $d \geqslant d_t$ we have that $p_{t+1}(A) = p_t(A)$ for all $A \in \mathcal{A}_d$.*

For the proof, we require the next technical result about the balancing vectors $\overline{\ell}_{t,d}$ computed by the algorithm.

Lemma 7. *If $\sigma_{t,0} = \ldots = \sigma_{t,d-1} = 1$ then:*

$$\sum_{i\in A} p_t(i) e^{-\eta \overline{\ell}_{t,d}(i)} = \sum_{i\in A} p_t(i) e^{-\eta \widetilde{\ell}_{t,d}(i)} \qquad \forall\ A \in \mathcal{A}_d\ ,$$

where $\widetilde{\ell}_{t,d} = \overline{\ell}_{t,0} + \sum_{h=0}^{d-1} \overline{\ell}_{t,h}$.

Proof. The proof proceeds by induction on d. For the base case $d = 0$, the claim follows trivially as $\overline{\ell}_{t,0} = \widetilde{\ell}_{t,0}$. Next, we assume the claim is true for some value of $d \geqslant 0$ and prove it for $d+1$. Pick any $A \in \mathcal{A}_{d+1}$ and write $A = A_1 \cup A_2$ where A_1, A_2 are disjoint sets from $\mathcal{A}_d$. Notice that the vector $\overline{\ell}_{t,d}$ is uniform over A_1 and A_2, namely $\overline{\ell}_{t,d}(i) = c_{A_1}$ for all $i \in A_1$ for some $c_{A_1} \geqslant 0$, and similarly $\overline{\ell}_{t,d}(i) = c_{A_2}$ for all $i \in A_2$ for some $c_{A_2} \geqslant 0$. Hence, we have

$$\begin{aligned}
\sum_{i\in A} p_t(i) e^{-\eta \widetilde{\ell}_{t,d+1}(i)} &= \sum_{i\in A} p_t(i) e^{-\eta \widetilde{\ell}_{t,d}(i)} e^{-\eta \overline{\ell}_{t,d}(i)} \\
&= e^{-\eta c_{A_1}} \sum_{i\in A_1} p_t(i) e^{-\eta \widetilde{\ell}_{t,d}(i)} + e^{-\eta c_{A_2}} \sum_{i\in A_2} p_t(i) e^{-\eta \widetilde{\ell}_{t,d}(i)} \\
&= e^{-\eta c_{A_1}} \sum_{i\in A_1} p_t(i) e^{-\eta \overline{\ell}_{t,d}(i)} + e^{-\eta c_{A_2}} \sum_{i\in A_2} p_t(i) e^{-\eta \overline{\ell}_{t,d}(i)} \\
&= \sum_{i\in A} p_t(i) e^{-\eta \overline{\ell}_{t,d}(i)} e^{-\eta \overline{\ell}_{t,d}(i)} \\
&= \sum_{i\in A} p_t(i) e^{-2\eta \overline{\ell}_{t,d}(i)}\ ,
\end{aligned}$$

where the third equality uses the induction hypothesis. On the other hand, by the recursive definition of $\overline{\ell}_{t,d+1}$ and the fact that $\overline{\ell}_{t,d+1}$ is uniform over A, we have

$$\sum_{i\in A} p_t(i)e^{-\eta\overline{\ell}_{t,d+1}(i)} = p_t(A)\sum_{i\in A}\frac{p_t(i)}{p_t(A)}e^{-\eta(1+\sigma_{t,d})\overline{\ell}_{t,d}(i)} = \sum_{i\in A} p_t(i)e^{-2\eta\overline{\ell}_{t,d}(i)} \ .$$

Combining both observations, we obtain

$$\sum_{i\in A} p_t(i)e^{-\eta\overline{\ell}_{t,d+1}(i)} = \sum_{i\in A} p_t(i)e^{-\eta\widetilde{\ell}_{t,d+1}(i)}$$

which concludes the inductive argument. ■

We can now prove Lemma 6.

Proof of Lemma 6. It is enough to prove that $p_{t+1}(A) = p_t(A)$ for all $A \in \mathcal{A}_{d_t}$, as each set in $\mathcal{A}_d$ for $d > d_t$ is a disjoint union of sets from $\mathcal{A}_{d_t}$.

Observe that if $i_t \in B_t$ (see Algorithm 1 for the definition of B_t) then $\widetilde{\ell}_t = 0$ and the claim is certainly true as $p_{t+1} = p_t$ in this case. Thus, we henceforth assume that $i_t \notin B_t$, in which case $\widetilde{\ell}_t = \widetilde{\ell}_{t,d_d} - \overline{\ell}_{t,d_d}$ where $\widetilde{\ell}_{t,d_t} = \overline{\ell}_{t,0} + \sum_{h=0}^{d_t-1}\overline{\ell}_{t,h}$ (recall Eq. (3)). Now, pick any $A \in \mathcal{A}_{d_t}$ and $j \in A$. Since $\overline{\ell}_{t,d_t}(i) = c_A$ for all $i \in A$ for some $c_A \geqslant 0$, and using Lemma 7 we obtain

$$e^{-\eta c_A} = \sum_{i\in A}\frac{p_t(i)}{p_t(A)}e^{-\eta\overline{\ell}_{t,d_t}(i)} = \sum_{i\in A}\frac{p_t(i)}{p_t(A)}e^{-\eta\widetilde{\ell}_{t,d_t}(i)} \ . \tag{4}$$

On the other hand, from $\widetilde{\ell}_t = \widetilde{\ell}_{t,d_t} - \overline{\ell}_{t,d_t}$ it follows that $e^{-\eta\widetilde{\ell}_t(i)} = e^{-\eta\widetilde{\ell}_{t,d_t}(i)}/e^{-\eta c_A}$ for all $i \in A$, and by Eq. (4) we have

$$\sum_{i\in A} p_t(i)e^{-\eta\widetilde{\ell}_t(i)} = \frac{\sum_{i\in A} p_t(i)e^{-\eta\widetilde{\ell}_{t,d_t}(i)}}{e^{-\eta c_A}} = p_t(A) \ .$$

In words, the multiplicative update does not change the probabilities of the sets in $\mathcal{A}_{d_t}$, hence $p_{t+1}(A) = p_t(A)$ for all $A \in \mathcal{A}_{d_t}$ as required. ■

4.2. Lazy sampling

Our next step is to show that the sampling scheme employed by Algorithm 1 is valid and gives rise to low movement costs on expectation. Specifically, we would like to show that in a certain sense, the action i_t on round t is distributed in expectation according to the distribution p_t, even though it is sampled from a conditional of p_t in a way that is highly correlated with the preceding action i_{t-1}. Furthermore, we will show that the correlations in the sampling scheme are designed in a way that the expected movement between consecutive actions is small. These properties are formalized in the following lemma.

Lemma 8. *For all t and $0 \leqslant d < D$ the following hold:*

- *for all $A \in \mathcal{A}_d$ we have*

$$\mathbb{E}\left[\frac{\mathbb{1}\{i_t \in A\}}{p_t(A)}\right] = 1 \ ; \tag{5}$$

- *with probability at least* $1 - 2^{-(d+1)}$*, we have that* $A_d(i_t) = A_d(i_{t-1})$.

Eq. (5) is central to our analysis below, and virtually all of our probabilistic arguments involving the random variables i_t and p_t will be based on this property. We remark that if we were to sample i_t directly from the distribution specified by p_t, then Eq. (5) would have been trivially true. However, the i_t are sampled from a conditional of p_t that might be very different from p_t itself; nevertheless, the lemma shows that Eq. (5) still continues to hold under the skewed sampling process.

Lemma 8 also implies the slow-movement property of the algorithm: at the high levels of the tree, where the subtrees are "wide", the actions i_t and i_{t-1} are very likely to belong to the same subtree. The probability of switching subtrees increases exponentially with the level in the tree: at the lower levels, where the subtrees are "narrow", subtree switches may occur more often as the movement cost incurred by such switches is low.

Proof of Lemma 8. The second statement is true since we pick $i_{t+1} \sim p_t(i \mid A_{d_t}(i_t))$, so that $A_d(i_{t+1}) \neq A_d(i_t)$ can occur only if $d < d_t$. This happens with probability $2^{-(d+1)}$.

Next, we show Eq. (5) by induction on t. For $t = 1$ the statement is true since $i_1 \sim p_1$. For the induction step, condition on d_t and fix any $d \geqslant d_t$ and $A \in \mathcal{A}_d$. By Lemma 6 we have that $p_t(A) = p_{t+1}(A)$. Also $i_t \in A$ if and only if $i_{t+1} \in A$, since $d \geqslant d_t$ implies that $i_t \in A$ if and only if $A_{d_t}(i_t) \subseteq A$ and $A_{d_t}(i_{t+1}) = A_{d_t}(i_t)$. Hence, we have

$$\mathbb{E}\left[\frac{\mathbb{1}\{i_{t+1} \in A\}}{p_{t+1}(A)} \;\middle|\; d_t\right] = \mathbb{E}\left[\frac{\mathbb{1}\{i_t \in A\}}{p_t(A)} \;\middle|\; d_t\right] = \mathbb{E}\left[\frac{\mathbb{1}\{i_t \in A\}}{p_t(A)}\right] = 1 \;, \tag{6}$$

where the last equality holds true since d_t depends solely on $\sigma_{t,0}, \ldots, \sigma_{t,D-1}$ which are independent of i_t and p_t (note that this equality then holds for *any* set A, regardless of the fact that $A \in \mathcal{A}_d$).

Next, we consider any $d < d_t$ and $A \in \mathcal{A}_d$. Let $A' \in \mathcal{A}_{d_t}$ be the subtree such that $A \subseteq A'$, and recall that $i_{t+1} \sim p_{t+1}(i \mid A_{d_t}(i_t))$. Hence,

$$\mathbb{E}\left[\frac{\mathbb{1}\{i_{t+1} \in A\}}{p_{t+1}(A)} \;\middle|\; i_t \in A', p_{t+1}, d_t\right] = \mathbb{E}\left[\frac{\mathbb{1}\{i_{t+1} \in A\}}{p_{t+1}(A \mid A')p_{t+1}(A')} \;\middle|\; i_t \in A', p_{t+1}, d_t\right] = \frac{1}{p_{t+1}(A')} \;. \tag{7}$$

Since $i_t \in A'$ implies that $i_{t+1} \in A'$, we have

$$\mathbb{E}\left[\frac{\mathbb{1}\{i_{t+1} \in A\}}{p_{t+1}(A)} \;\middle|\; d_t, p_{t+1}\right] = \mathbb{E}\left[\mathbb{1}\{i_{t+1} \in A'\} \cdot \mathbb{E}\left[\frac{\mathbb{1}\{i_{t+1} \in A\}}{p_{t+1}(A)} \;\middle|\; i_t \in A', p_{t+1}, d_t\right] \;\middle|\; d_t, p_{t+1}\right] \;. \tag{8}$$

Taking Eqs. (7) and (8) together and taking the expectation over p_{t+1}, we obtain that for every $d < d_t$:

$$\mathbb{E}\left[\frac{\mathbb{1}\{i_{t+1} \in A\}}{p_{t+1}(A)} \;\middle|\; d_t\right] = \mathbb{E}\left[\frac{\mathbb{1}\{i_{t+1} \in A'\}}{p_{t+1}(A')} \;\middle|\; d_t\right] = 1 \;,$$

where last equality follows from Eq. (6) as $A' \in \mathcal{A}_{d_t}$.

To conclude, we showed that for all d we have:

$$\mathbb{E}\left[\frac{\mathbb{1}\{i_{t+1} \in A\}}{p_{t+1}(A)} \;\middle|\; d_t\right] = 1 \;.$$

Taking the expectation over d_t, we obtain the desired result. ■

4.3. Bounding the bias and variance

Next, we turn to bound the variance of the loss estimates $\widetilde{\ell}_t$ and the bias of their expectations from the true loss vectors. These bounds would become useful for controlling the expected regret of the underlying multiplicative updates scheme.

We begin with analyzing the bias of our estimator. The following lemma shows that our estimates are "optimistic", in the sense that they always bound the true losses from below, yet they do not overly underestimate the losses incurred by the algorithm. The proof is somewhat involved, as a result of the "bad events" B_t under which the estimated loss vectors $\widetilde{\ell}_t$ are being zeroed-out, thereby introducing biases into the estimation.

Lemma 9. *For all t, we have $\mathbb{E}[\widetilde{\ell}_t(i)] \leqslant \ell_t(i)$ and $\mathbb{E}[\ell_t(i_t)] \leqslant \mathbb{E}[p_t \cdot \widetilde{\ell}_t] + \eta k \log_2 k$.*

Proof. Observe that, by Eq. (5) of Lemma 8,

$$\mathbb{E}[\overline{\ell}_{t,0}(i)] = \ell_t(i)\,\mathbb{E}\left[\frac{\mathbb{1}\{i_t = i\}}{p_t(i)}\right] = \ell_t(i) \;.$$

We now prove that $\mathbb{E}[\widetilde{\ell}_{t,0}(i)] \leqslant \mathbb{E}[\overline{\ell}_{t,0}(i)]$ for all i, which would imply the first claim. By construction we have $\mathbb{E}[\widetilde{\ell}_t(i) \mid i_t \in B_t] = 0 \leqslant \mathbb{E}[\overline{\ell}_{t,0}(i) \mid i_t \in B_t]$. Also, since $\mathbb{E}[\sigma_{t,d}] = 0$ and $\sigma_{t,d}$ is independent of i_t and $\overline{\ell}_{t,d}$ (the latter only depends on $\sigma_{t,0}, \ldots, \sigma_{t,d-1}$), we have

$$\mathbb{E}[\widetilde{\ell}_t \mid i_t \notin B_t] = \mathbb{E}[\overline{\ell}_{t,0} \mid i_t \notin B_t] + \sum_{d=0}^{D-1} \mathbb{E}[\sigma_{t,d}]\,\mathbb{E}[\overline{\ell}_{t,d} \mid i_t \notin B_t] = \mathbb{E}[\overline{\ell}_{t,0} \mid i_t \notin B_t] \;. \tag{9}$$

Together, we obtain $\mathbb{E}[\widetilde{\ell}_{t,0}(i)] \leqslant \mathbb{E}[\overline{\ell}_{t,0}(i)]$ as required.

Next, to bound $\mathbb{E}[\ell_t(i_t)]$ observe that $\mathbb{E}[p_t \cdot \widetilde{\ell}_t \mid i_t \in B_t] = 0$ and, similarly to Eq. (9),

$$\mathbb{E}[p_t \cdot \widetilde{\ell}_t \mid i_t \notin B_t] = \mathbb{E}[p_t \cdot \overline{\ell}_{t,0} \mid i_t \notin B_t] = \mathbb{E}[\ell_t(i_t) \mid i_t \notin B_t] \;.$$

Denote $\beta_t = \mathbb{P}\left[i_t \in B_t\right]$. Then

$$\begin{aligned}
\mathbb{E}[\ell_t(i_t)] &= \beta_t \mathbb{E}[\ell_t(i_t) \mid i_t \in B_t] + (1-\beta_t)\mathbb{E}[\ell_t(i_t) \mid i_t \notin B_t] \\
&\leqslant \beta_t + (1-\beta_t)\mathbb{E}[p_t \cdot \widetilde{\ell}_t \mid i_t \notin B_t] \\
&= \beta_t + \mathbb{E}[p_t \cdot \widetilde{\ell}_t] \;,
\end{aligned}$$

where for the inequality we used the fact that $\ell_t(i_t) \leqslant 1$.

To complete the proof, we have to show that $\beta_t \leqslant \eta k \log_2 k$. To this end, write

$$\mathbb{P}[i_t \in B_t] \leqslant \sum_{d=0}^{D-1} \mathbb{P}[p_t(A_d(i_t)) < 2^d \eta] \;.$$

Using Eq. (5) to write

$$\mathbb{E}\left[\frac{1}{p_t(A_d(i_t))}\right] = \sum_{i=1}^{k} \frac{1}{|A_d(i)|}\mathbb{E}\left[\frac{\mathbb{1}\{i_t \in A_d(i)\}}{p_t(A_d(i))}\right] = \sum_{i=1}^{k} \frac{1}{|A_d(i)|} = |\mathcal{A}_d| = \frac{k}{2^d}$$

together with Markov's inequality, we obtain

$$\mathbb{P}\Big[p_t(A_d(i_t)) < 2^d \eta\Big] = \mathbb{P}\left[\frac{1}{p_t(A_d(i_t))} > \frac{1}{2^d \eta}\right] \leqslant \frac{k}{2^d} \cdot 2^d \eta = k\eta \;.$$

We conclude that $\beta_t = \mathbb{P}[i_t \in B_t] \leqslant \eta k \log_2 k$, as required. ∎

Our next step is to bound the relevant variance term of the estimator $\widetilde{\ell}_t$.

Lemma 10. *For all t, we have $\mathbb{E}[p_t \cdot \widetilde{\ell}_t^2] \leqslant 2k \log_2 k$.*

Proof. Observe that

$$\widetilde{\ell}_t^2(i) \leqslant \left(\overline{\ell}_{t,0}(i) + \sum_{d=0}^{D-1} \sigma_{t,d} \overline{\ell}_{t,d}(i) \right)^2 .$$

Since $\mathbb{E}[\sigma_{t,d}] = 0$ and $\mathbb{E}[\sigma_{t,d}\sigma_{t,d'}] = 0$ for all $d \neq d'$, we have for all i that

$$\mathbb{E}[\widetilde{\ell}_t^2(i)] = \mathbb{E}[\widetilde{\ell}_{t,0}^2(i)] + \sum_{d=0}^{D-1} \mathbb{E}[\overline{\ell}_{t,d}^2(i)] \leqslant 2 \sum_{d=0}^{D-1} \mathbb{E}[\overline{\ell}_{t,d}^2(i)] . \tag{10}$$

On the other hand, for all d we have by Lemma 5 that

$$\begin{aligned} p_t \cdot \overline{\ell}_{t,d}^2 &\leqslant \frac{\sum_{i=1}^k p_t(i) \mathbb{1}\{i_t \in A_d(i)\}}{p_t(A_d(i_t))^2} \prod_{h=0}^{d-1} (1 + \sigma_{t,h})^2 \\ &= \frac{1}{p_t(A_d(i_t))} \prod_{h=0}^{d-1} (1 + \sigma_{t,h})^2 \\ &= \sum_{i=1}^k \frac{1}{|A_d(i)|} \frac{\mathbb{1}\{i_t \in A_d(i)\}}{p_t(A_d(i))} \prod_{h=0}^{d-1} (1 + \sigma_{t,h})^2 . \end{aligned}$$

Since i_t is independent of the $\sigma_{t,h}$, and recalling Eq. (5), we get

$$\mathbb{E}[p_t \cdot \overline{\ell}_{t,d}^2] \leqslant \sum_{i=1}^k \frac{1}{|A_d(i)|} \mathbb{E}\left[\frac{\mathbb{1}\{i_t \in A_d(i)\}}{p_t(A_d(i))} \right] \prod_{h=0}^{d-1} \mathbb{E}[(1 + \sigma_{t,h})^2] = \sum_{i=1}^k \frac{2^d}{|A_d(i)|} = 2^d |\mathcal{A}_d| = k .$$

Together with Eq. (10), this gives

$$\mathbb{E}[p_t \cdot \widetilde{\ell}_t^2] \leqslant 2 \sum_{d=0}^{D-1} \mathbb{E}[p_t \cdot \overline{\ell}_{t,d}^2] \leqslant 2k \log_2 k. \qquad \blacksquare$$

4.4. Concluding the proof

To conclude the proof and obtain a regret bound, we will use the following well-known second-order regret bound for the multiplicative weights (MW) method, essentially due to Cesa-Bianchi et al. (2007) (see also Alon et al. (2015) for the version given here). For completeness, we give a proof of this bound in Section 4.5 below.

Lemma 11 (Second-order regret bound for MW)**.** *Let $\eta > 0$ and let $c_1, \ldots, c_T \in \mathbb{R}^k$ be real vectors such that $c_t(i) \geqslant -1/\eta$ for all t and i. Consider a sequence of probability vectors $q_1, \ldots, q_T \in \Delta_k$ defined by $q_1 = (\frac{1}{k}, \ldots, \frac{1}{k})$, and for all $t > 1$:*

$$q_{t+1}(i) = \frac{q_t(i)\, e^{-\eta c_t(i)}}{\sum_{j=1}^k q_t(j)\, e^{-\eta c_t(j)}} \qquad \forall\ i \in [k] .$$

Then, for all $i^* \in [k]$ we have that

$$\sum_{t=1}^{T} q_t \cdot c_t - \sum_{t=1}^{T} c_t(i^*) \leqslant \frac{\log k}{\eta} + \eta \sum_{t=1}^{T} q_t \cdot c_t^2 \ .$$

We now have all we need in order to prove our main result.

Proof of Theorem 4. First, we bound the expected movement cost. Lemma 8 says that with probability at least $1 - 2^{-(d+1)}$, the actions i_t and i_{t-1} belong to the same subtree on level d of the tree, which means that $\Delta(i_t, i_{t-1}) \leqslant 2^d/k$ with the same probability. Hence,

$$\mathbb{E}[\Delta(i_t, i_{t-1})] \leqslant \sum_{d=0}^{D-1} \frac{2^d}{k} \mathbb{P}\left[\Delta(i_t, i_{t-1}) > \frac{2^d}{k}\right] \leqslant \sum_{d=0}^{D-1} \frac{1}{2k} = \frac{\log_2 k}{2k} \ ,$$

and the cumulative movement cost is then $O((T/k)\log k)$.

We turn to analyze the cumulative loss of the algorithm. We begin by observing that $\widetilde{\ell}_t(i) \geqslant -1/\eta$ for all t and i. To see this, notice that $\widetilde{\ell}_t = 0$ unless $i_t \notin B_t$, in which case we have, by Lemma 5 and the definition of B_t,

$$0 \leqslant \overline{\ell}_{t,d}(i) \leqslant \frac{2^d}{p_t(A_d(i_t))} \leqslant \frac{1}{\eta} \qquad \forall\ 0 \leqslant d < D \ ,$$

and since $\widetilde{\ell}_t$ has the form $\widetilde{\ell}_t = \overline{\ell}_{t,0} + \sum_{h=0}^{d_t-1} \overline{\ell}_{t,h} - \overline{\ell}_{t,d_t}$ (recall Eq. (3)), we see that $\widetilde{\ell}_t(i) \geqslant -1/\eta$. Hence, we can use second-order bound of Lemma 11 on the vectors $\widetilde{\ell}_t$ to obtain

$$\sum_{t=1}^{T} p_t \cdot \widetilde{\ell}_t - \sum_{t=1}^{T} \widetilde{\ell}_t(i^*) \leqslant \frac{\log k}{\eta} + \eta \sum_{t=1}^{T} p_t \cdot \widetilde{\ell}_t^2$$

for any fixed $i^* \in [k]$. Taking expectations and using Lemmas 9 and 10, we have

$$\mathbb{E}\left[\sum_{t=1}^{T} \ell_t(i_t)\right] - \sum_{t=1}^{T} \ell_t(i^*) \leqslant \frac{\log_2 k}{\eta} + 2\eta T k \log_2 k \ .$$

Choosing $\eta = 1/\sqrt{Tk}$, we get a regret bound of $O(\sqrt{Tk}\log k)$. ∎

4.5. Additional technical proofs

Here we give a proof of our technical lemma bounding the magnitude of the balancing terms $\overline{\ell}_{t,d}$.

Proof of Lemma 5. We will prove the claim by induction on d. For the base case $d = 0$, Eq. (2) follows directly from our definitions and the fact that $0 \leqslant \ell_t(i) \leqslant 1$ for all i. Next, we prove that Eq. (2) holds for some d assuming it hold for all $d' < d$. Since $(1 + \sigma_{t,d-1})\overline{\ell}_{t,d-1}(i) \geqslant 0$ for all i by the induction hypothesis, the recursive definition of $\overline{\ell}_{t,d}$ implies that

$$\overline{\ell}_{t,d}(i) \geqslant -\frac{1}{\eta} \log\left(\sum_{j \in A_d(i)} \frac{p_t(j)}{p_t(A_d(j))}\right) = 0 \ .$$

Furthermore, the definition of $\overline{\ell}_{t,d}$ together with the convexity of $-\log x$ and Jensen's inequality give

$$
\begin{aligned}
\overline{\ell}_{t,d}(i) &\leqslant (1+\sigma_{d-1}) \sum_{j \in A_d(i)} \frac{p_t(j)}{p_t(A_d(j))} \overline{\ell}_{t,d-1}(j) \\
&\leqslant \frac{\mathbb{1}\{i_t \in A_d(i)\}}{p_t(A_d(i))} \sum_{j \in A_{d-1}(i)} \frac{p_t(j)}{p_t(A_{d-1}(j))} \prod_{h=0}^{d-1}(1+\sigma_{t,h}) \\
&= \frac{\mathbb{1}\{i_t \in A_d(i)\}}{p_t(A_d(i))} \prod_{h=1}^{d-1}(1+\sigma_{t,h}) \ ,
\end{aligned}
$$

where in the second inequality we used the induction hypothesis. This concludes the inductive argument. ∎

Finally, for completeness, we give a proof of Lemma 11 being central to our regret analysis.

Proof of Lemma 11. The proof follows the standard analysis of exponential weighting schemes: let $w_t(i) = \exp\big(-\eta \sum_{s=1}^{t-1} c_s(i)\big)$ and let $W_t = \sum_{i \in V} w_t(i)$. Then $q_t(i) = w_t(i)/W_t$ and we can write

$$
\begin{aligned}
\frac{W_{t+1}}{W_t} &= \sum_{i=1}^{k} \frac{w_{t+1}(i)}{W_t} \\
&= \sum_{i=1}^{k} \frac{w_t(i)\, \exp\big(-\eta\, c_t(i)\big)}{W_t} \\
&= \sum_{i=1}^{k} q_t(i)\, \exp\big(-\eta\, c_t(i)\big) \\
&\leqslant \sum_{i=1}^{k} q_t(i)\, \big(1 - \eta c_t(i) + \eta^2 c_t(i)^2\big) \\
&= 1 - \eta \sum_{i=1}^{k} q_t(i) c_t(i) + \eta^2 \sum_{i=1}^{k} q_t(i) c_t(i)^2 \ ,
\end{aligned}
$$

where the inequality uses the inequality $e^x \leqslant 1 + x + x^2$ valid for $x \leqslant 1$. Taking logarithms, using $\log(1-x) \leqslant -x$ for all $x \leqslant 1$, and summing over $t = 1, \ldots, T$ yields

$$
\log \frac{W_{T+1}}{W_1} \leqslant \sum_{t=1}^{T} \sum_{i=1}^{k} \big(-\eta\, q_t(i) c_t(i) + \eta^2\, q_t(i) c_t(i)^2\big) \ .
$$

Moreover, for any fixed action i^*, we also have

$$
\log \frac{W_{T+1}}{W_1} \geqslant \log \frac{w_{T+1}(k)}{W_1} = -\eta \sum_{t=1}^{T} c_t(i^*) - \log k \ .
$$

Putting together and rearranging gives the result. ∎

4.6. Learning Continuum–Arm Bandit with Lipschitz Loss Functions

In this section we turn to show how to reduce the problem of learning Lipschitz functions to MAB with tree-metric movement costs. Specifically we aim at proving Theorem 2, which follows directly from the following statement,

Theorem 12. *Set $k = L^{2/3}T^{1/3}$ and $\eta = 1/\sqrt{kT}$. Consider a procedure that receives actions from Algorithm 1 and returns as feedback $f_t(\frac{i_t}{k})$ then for every sequence of L-Lipschitz loss functions $f_1, \dots, f_T$ and an L-Lipschitz metric Δ, we have that:*

$$\mathit{Regret}_{\mathsf{MC}}(f_{1:T}, \Delta) = \widetilde{O}\big(L^{1/3}T^{2/3}\big) \ .$$

In particular, the result holds for $L \geqslant 1$ and $\Delta(x_t, x_{t+1}) = |x_t - x_{t+1}|$.

Proof. First note that for every $x^* \in [0,1]$ we can find $x = \{\frac{1}{k}, \frac{2}{k}, \dots, 1\}$ such that $f_t(x) - f_t(x^*) \leqslant L/k = L^{1/3}T^{-1/3}$, hence

$$\sum_{t=1}^{T} \big(f_t(x) - f_t(x^*)\big) = L^{1/3}T^{2/3}.$$

Therefore if we can show that the regret against every $x^* \in \{\frac{1}{k}, \frac{2}{k}, \dots 1\}$ is bounded by $O(L^{1/3}T^{2/3})$ we obtain that the same regret bound is true for every $x \in [0,1]$.

Next, we apply Algorithm 1 on the a fully balanced tree where we associate with the leaves $\{1, \dots, k\}$ the actions $\{\frac{1}{k}, \frac{2}{k} \dots, 1\}$. One can then show that $\frac{|i-j|}{k} \leqslant \Delta_{\mathcal{T}}(i,j)$. We then obtain by Theorem 4 that for every $x \in \{\frac{1}{k}, \frac{2}{k} \dots, 1\}$:

$$\mathbb{E}\left[\sum_{t=1}^{T} f_t(x_t)\right] - \min_{x} \sum_{t=1}^{T} f_t(x) = O(\eta k T) = \widetilde{O}(L^{1/3}T^{2/3}) \ .$$

As to the second term in the regret we obtain that

$$\mathbb{E}\left[\sum_{t=1}^{T} \Delta(x_t, x_{t+1})\right] \leqslant L \sum_{t=1}^{T} |x_t - x_{t+1}| \leqslant \mathbb{E}\left[L \sum_{t=1}^{T} \Delta_{\mathcal{T}}(i_t, i_{t+1})\right] = \widetilde{O}\left(L\frac{T}{k}\right) = \widetilde{O}(L^{1/3}T^{2/3}) \ .$$

Taken together we obtain that

$$\mathbb{E}\left[\sum_{t=1}^{T} f_t(x_t) + \sum_{t=1}^{T} \Delta(x_t, x_{t+1})\right] - \min_{x \in \{\frac{1}{k}, \dots, 1\}} \sum_{t=1}^{T} f_t(x) = \widetilde{O}(L^{1/3}T^{2/3}) \ . \qquad \blacksquare$$

5. Online Pricing with Patient Buyers

In this section we present our reduction of adaptive pricing with patient buyers to a MAB with movement costs.

The reduction is presented in Algorithm 2 and uses our algorithm for MAB with movement costs (Algorithm 1) as a black-box. The algorithm divides the time interval T into $\overline{\tau}$ blocks and updates the price on $\overline{T} = T/\overline{\tau}$ time steps. At each time step t the algorithm publishes a fixed price for the whole block of $\overline{\tau}$ consecutive days. Then, as feedback, the algorithm receives the mean revenue for those days, which we denote by

$$r'_t = \frac{1}{\overline{\tau}} \sum_{k=(t-1)\overline{\tau}+1}^{t\overline{\tau}} \mathbf{b}_k(\rho_k, \ldots, \rho_{k+\overline{\tau}}) \ .$$

Thus, we can consider the algorithm as an online algorithm over $\overline{T}$ rounds: where at each round t the algorithm announces a fixed action ρ'_{t+1} (the price for the next $\overline{\tau}$ days) and receives at the end of the round as feedback r'_t. Note that prices are always announced $\overline{\tau}$ days in advance, as required. Dividing the horizon into $T/2\overline{\tau}$ blocks ensures that buyers see at most two different prices. This in turns lead to a reduction to the case $\overline{\tau} = 1$.

Next, as discussed briefly in Section 3, the main difficulty in reducing the adaptive pricing problem to MAB, which Algorithm 2 overcomes, is in that the feedback function is not only a function of the current posted price (which is in fact the price tomorrow) but also of past prices. For example, for $\overline{\tau} = 1$ the revenue at time t is a function of ρ_t and ρ_{t+1}, where only ρ_{t+1} needs be posted at time t. Algorithm 2 overcomes this issue by employing techniques from Dekel et al. (2014b) for handling adaptive feedback. The tools developed there allow regret minimization when feedback is taken only in time steps when the price is fixed for a period of time.

The algorithm draws $\beta_1, \ldots, \beta_{\overline{T}}$ unbiased Bernoulli random variables, and this sequence determines the switches in prices and updates. The algorithm posts a new price only on rounds where $\beta_t = 0$ and $\beta_{t+1} = 1$, and invoke the update rule of Algorithm 1 only on rounds where $\beta_{t+1} = 0$ and $\beta_{t+2} = 1$. Note that these two events never co-occur, and further the algorithm exploits the feedback only on days prior to a switch, thus guaranteeing that the feedback is always on days when prices are fixed throughout the present and future block.

Parameters: horizon T, and maximal patience $\overline{\tau}$
Initialize, $\overline{T} = T/(2\overline{\tau})$, $k = \overline{T}^{1/3}$, $\eta = 2/\sqrt{\overline{T}k}$
Initialize an instance B of SMB(k, η)
Draw i.i.d. unbiased Bernoulli r.v. $\beta_0, \ldots, \beta_{\overline{T}}$
Sample $i_1 \sim B$, set $\rho'_1 = i_1/k$
Announce prices $\rho_1 = \rho_2 = \ldots, p_{\overline{\tau}} = \rho'_1$
For $t = 1, \ldots, \overline{T}$

(1) If $\beta_t = 0$ and $\beta_{t+1} = 1$, sample $i_{t+1} \sim B$; otherwise set $i_{t+1} = i_t$
(2) Set $\rho'_{T+1} = i_{t+1}/k$ and announce prices: $\rho_{t\overline{\tau}+1} = \cdots = \rho_{(t+1)\overline{\tau}} = \rho'_{t+1}$
(3) Collect revenues $r_{(t-1)\overline{\tau}+1}, \ldots, r_{t\overline{\tau}}$ and set

$$r'_t(\rho'_t) = \frac{1}{\overline{\tau}} \sum_{k=(t-1)\overline{\tau}+1}^{t\overline{\tau}} r_k$$

(4) If $\beta_{t+1} = 0, \beta_{t+2} = 1$, update B with feedback $f_t = 1 - r'_t(\rho'_t)$

Algorithm 2: Adaptive pricing with patient buyers.

Relying on these techniques, we construct an algorithm that produces a sequence of prices with low regret *if* each buyer $\mathbf{b}_t$ would observe price ρ_t. However, in our setting, a buyer may buy at a consecutive time steps; the additional cost we suffer is bounded by the potential cost of switching to lower prices, namely, by the movement cost of the algorithm.

The remainder of the section is devoted to proving Theorem 3. We begin by establishing additional notation required for the proof. We will denote the expected revenue from the buyers at each block as follows:

$$\overline{\mathbf{b}}_t(\rho'_t, \rho'_{t+1}) = \frac{1}{\overline{\tau}} \sum_{k=t\overline{\tau}+1}^{(t+1)\overline{\tau}} \mathbf{b}_k(\rho_k, \ldots, \rho_{k+\tau_t}) \ .$$

Note that since the blocks are of size $\overline{\tau}$, each buyer can see at most prices that are published on the next block, hence $\rho_{k+\tau_t}$ either equals ρ'_t or ρ'_{t+1}. In turn, this means that the expected revenue is indeed a function of ρ'_t and ρ'_{t+1} alone.

We will further denote the expected revenue from buyers if they observe only the price at time of arrival as follows:

$$\overline{\mathbf{b}}_t(\rho'_t) = \frac{1}{\overline{\tau}} \sum_{k=t\overline{\tau}+1}^{(t+1)\overline{\tau}} \mathbf{b}_k(\rho'_t, \ldots, \rho'_t) \ .$$

First, we are estimating the performance on the subsequence of rounds where the algorithm exploits the received feedback.

Lemma 13. *Let $\beta_1, \ldots, \beta_T$ be a sequence of unbiased Bernoulli random variables, denote*

$$S = \{t \in [\overline{T}] \ : \ \beta_{t+1} = 0, \beta_{t+2} = 1\},$$

and denote the elements of S in increasing order $S = \{t_{s_1} \leqslant t_{s_2}, \ldots, \leqslant t_{s_{|S|}}\}$. For any price $\rho^* \in \{\frac{1}{k}, \frac{2}{k}, \ldots, 1\}$, Algorithm 2 enjoys the following guarantee:

$$\mathbb{E}\left[\sum_{t\in S} \overline{\mathbf{b}}_t(\rho^*) - \overline{\mathbf{b}}_t(\rho'_t)\right] = \widetilde{O}(\overline{T}^{2/3}) \ ,$$

and

$$\mathbb{E}\left[\sum_{s=1}^{|S|} |\rho'_{t_s} - \rho'_{t_{s+1}}|\right] = \widetilde{O}(\overline{T}^{2/3}) \ .$$

Proof. For each sequence of buyers $\mathbf{b}_1, \ldots, \mathbf{b}_T$, define a sequence of loss functions $\ell_1 \ldots, \ell_{\overline{T}}$ according to:

$$\ell_t(i) = 1 - \overline{\mathbf{b}}_t\left(\frac{i}{k}\right) \ .$$

First note that for every $t \in S$ we have $\rho'_t = \rho'_{t+1}$. The algorithm, in turn, announces the same price ρ'_t for all days: $\{(t-1)\overline{\tau}+1, \ldots, (t+1)\overline{\tau}\}$, hence the revenue obtained from buyer $\mathbf{b}_k$ for every $(t-1)\overline{\tau}+1 \leqslant k \leqslant t\overline{\tau}$ is given by $\mathbf{b}_t(\rho'_t, \rho'_t)$. Hence, the feedback used to update the algorithm B at round t is

$$f_t = 1 - r'_t = 1 - \frac{1}{\overline{\tau}} \sum_{k=(t-1)\overline{\tau}+1}^{t\overline{\tau}} \mathbf{b}_k(\rho_k, \ldots, \rho_{k+\overline{\tau}}) = 1 - \sum_{k=(t-1)\overline{\tau}+1}^{t\overline{\tau}} \frac{1}{\overline{\tau}} \mathbf{b}_k(\rho'_t) = \ell_t(i_t) \ .$$

In words, we have shown that at every step $t \in S$, Algorithm 2 receive action i_t and return to Algorithm 1 as feedback $\ell_t(i_t)$. Thus Algorithm 2 applies Algorithm 1 on the sequence of losses $\{\ell_t\}_{t \in S}$. As a corollary we have that:

$$\mathbb{E}\left[\sum_{t \in S} \overline{\mathbf{b}}_t(\rho^*) - \overline{\mathbf{b}}_t(\rho'_t) \;\middle|\; S \right] = \mathbb{E}\left[\sum_{t \in S} \ell_t(i^*) - \ell_t(i_t) \;\middle|\; S \right] = O(\eta k |S|) \ .$$

Taking expectation over S and noting $\mathbb{E}[|S|] = \frac{1}{4}\overline{T}$ we get that

$$\mathbb{E}\left[\sum_{t \in S} \overline{\mathbf{b}}_t(\rho^*) - \overline{\mathbf{b}}_t(\rho'_t,) \right] = O(\overline{T}^{2/3}) \ .$$

As in Section 4.6, note that if we associate with the prices the corresponding actions on the tree we obtain that $|\rho'_t - \rho'_{t+1}| \leqslant \Delta_{\mathcal{T}}(i_t, i_{t+1})$ hence we obtain as a second guarantee that the movement cost of the algorithm is given by

$$\mathbb{E}\left[\sum_{s=1}^{|S|} |\rho'_{t_s} - \rho'_{t_{s-1}}| \;\middle|\; S \right] = \mathbb{E}\left[\sum_{s=1}^{|S|} \tfrac{1}{k} |i_{t_s} - i_{t_{s-1}}| \;\middle|\; S \right] \leqslant \mathbb{E}\left[\sum_{s=1}^{|S|} \tfrac{1}{k} \Delta(i_{t_s}, i_{t_{s-1}}) \;\middle|\; S \right] = O\left(\tfrac{1}{k}|S|\right) \ .$$

Again taking expectation over S we get that

$$\mathbb{E}\left[\sum_{s=1}^{|S|} |\rho'_{t_s} - \rho'_{t_{s-1}}| \right] = \widetilde{O}\left(\tfrac{1}{k}\overline{T}\right) \ . \qquad \blacksquare$$

Next, we upper bound the regret over the expected regret over the blocks of buyers, $\bar{\mathbf{b}}_t$:

Lemma 14. *For every $\rho^* \in \{\frac{1}{k}, \frac{2}{k}, \ldots, 1\}$ we have that*

$$\mathbb{E}\left[\sum_{t=1}^{\overline{T}} \overline{\mathbf{b}}_t(\rho^*) - \overline{\mathbf{b}}_t(\rho'_t, \rho'_{t+1}) \right] \leqslant 4\mathbb{E}\left[\sum_{t \in S} \overline{\mathbf{b}}_t(\rho^*) - \overline{\mathbf{b}}_t(\rho'_t) \right] + \mathbb{E}\left[\sum_{s=1}^{|S|} |\rho'_{t_s} - \rho'_{t_{s-1}}| \right] \ .$$

Proof. First note that for every ρ^* we have

$$\mathbb{E}\left[\sum_{t \in S} \overline{\mathbf{b}}_t(\rho^*) \right] = \mathbb{E}\left[\sum_{t=1}^{T} \overline{\mathbf{b}}_t(\rho^*) \beta_{t+2}(1 - \beta_{t+1}) \right] \ .$$

Since the Bernoulli random variables are independent of $\mathbf{b}_t$ and ρ^* we get that

$$\mathbb{E}\left[\sum_{t \in S} \overline{\mathbf{b}}_t(\rho^*) \right] = \mathbb{E}\left[\sum_{t=1}^{T} \overline{\mathbf{b}}_t(\rho^*) \beta_{t+2}(1 - \beta_{t+1}) \right] = \frac{1}{4}\mathbb{E}\left[\sum_{t=1}^{T} \overline{\mathbf{b}}_t(\rho^*) \right] \ . \tag{11}$$

Similarly we have that

$$\mathbb{E}\left[\sum_{t\in S}\overline{\mathbf{b}}_t(\rho'_t)\right] = \mathbb{E}\left[\sum_{t=1}^{\overline{T}}\overline{\mathbf{b}}_t(\rho'_t)\beta_{t+2}(1-\beta_{t+1})\right] = \frac{1}{4}\mathbb{E}\left[\sum_{t=1}^{\overline{T}}\overline{\mathbf{b}}_t(\rho'_t)\right] ,$$

where the equality holds since ρ'_t is independent of β_{t+1} and β_{t+2}. We can bound $\mathbf{b}_t(\rho'_t, \rho'_{t+1}) \geqslant \mathbf{b}_t(\rho'_t, \rho'_t) - |\rho'_t - \rho'_{t+1}|$. Hence $\overline{\mathbf{b}}_t(\rho'_t, \rho'_{t+1}) \geqslant \overline{\mathbf{b}}(\rho'_t) - |\rho'_t - \rho'_{t+1}|$, and we obtain:

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{\overline{T}}\overline{\mathbf{b}}_t(\rho'_t, \rho'_{t+1})\right] &\geqslant \mathbb{E}\left[\sum_{t=1}^{\overline{T}}\overline{\mathbf{b}}_t(\rho'_t) - |\rho'_t - \rho'_{t+1}|\right] \\
&= 4\mathbb{E}\left[\sum_{t\in S}\overline{\mathbf{b}}_t(\rho'_t)\right] - \sum_{t=1}^{\overline{T}}\mathbb{E}\big[|\rho'_t - \rho'_{t+1}|\big] \\
&= 4\mathbb{E}\left[\sum_{t\in S}\overline{\mathbf{b}}_t(\rho'_t)\right] - \mathbb{E}\left[\sum_{t=s}^{|S|}|\rho'_{t_s} - \rho'_{t_{s-1}}|\right] , \qquad (12)
\end{aligned}$$

where last equality is true since, we have that $\rho'_t = \rho'_{t+1}$ unless $\rho'_{t-1} \in S$ in which case we have that $\rho'_{t-1} = \rho'_t = \rho'_{t_s}$ for some s and $\rho'_{t+1} = \rho'_{t_{s+1}}$. Taken together with Eqs. (11) and (12) we obtain the desired result. ■

We are now ready to prove the main result of this section.

Proof of Theorem 3. First, for any $\rho \in \{\frac{1}{k}, \ldots, 1\}$, by employing Lemma 14 we have the following:

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{T}\mathbf{b}_t(\rho,\ldots,\rho) - \mathbf{b}_t(\rho_t,\ldots,\rho_{t+\overline{\tau}})\right] &= \sum_{t'=1}^{\overline{T}}\sum_{t=(t'-1)\overline{\tau}+1}^{t'\overline{\tau}}\big(\mathbf{b}'_t(\rho,\ldots,\rho) - \mathbf{b}_t(\rho_t,\ldots,\rho_{t+\overline{\tau}})\big) \\
&= \overline{\tau}\mathbb{E}\left[\sum_{t=1}^{\overline{T}}\overline{\mathbf{b}}_t(\rho) - \overline{\mathbf{b}}_t(\rho'_t, \rho'_{t+1})\right] \\
&\leqslant \frac{\overline{\tau}}{4}\mathbb{E}\left[\sum_{t\in S}\overline{\mathbf{b}}_t(\rho) - \overline{\mathbf{b}}_t(\rho'_t)\right] + \overline{\tau}\mathbb{E}\left[\sum_{s=1}^{|S|}|\rho'_{t_s} - \rho'_{t_{s-1}}|\right] .
\end{aligned}$$

Next, for any $\rho^* \in [0,1]$ there exist $\rho \in \{\frac{1}{k}, \ldots, 1\}$ such that $\rho^* > \rho$ and $\mathbf{b}_t(\rho^*, \ldots, \rho^*) < \mathbf{b}_t(\rho, \ldots, \rho) + \frac{1}{k}$. Hence, for every $\rho^* \in [0,1]$ we obtain that

$$\begin{aligned}
&\sum_{t=1}^{T}\mathbf{b}_t(\rho^*,\ldots,\rho^*) - \mathbb{E}\left[\sum_{t=1}^{T}\mathbf{b}_t(\rho_t,\ldots\rho_{t+\overline{\tau}})\right] \\
&\qquad\leqslant \frac{\overline{\tau}}{4}\mathbb{E}\left[\sum_{t\in S}\overline{\mathbf{b}}_t(\rho) - \overline{\mathbf{b}}_t(\rho'_t)\right] + \overline{\tau}\mathbb{E}\left[\sum_{s=1}^{|S|}|\rho'_{t_s} - \rho'_{t_{s-1}}|\right] + O(\tfrac{T}{k}) .
\end{aligned}$$

By Lemma 13 we now obtain

$$\sum_{t=1}^{T} \mathbf{b}_t(\rho^*, \ldots, \rho^*) - \mathbb{E}\left[\sum_{t=1}^{T} \mathbf{b}_t(\rho_t, \ldots, \rho_{t+\overline{\tau}})\right] = O\left(\sqrt{\overline{\tau} k T} + \frac{\overline{\tau} T}{k} + \frac{T}{k}\right) = O(\overline{\tau}^{1/3} T^{2/3}) ,$$

and using our choice of k gives the result. ■

Acknowledgments

RL is supported by funding from the Eric and Wendy Schmidt Fund for Strategic Innovation. YM is supported in part by a grant from the Israel Science Foundation, a grant from the United States-Israel Binational Science Foundation (BSF), and the Israeli Centers of Research Excellence (I-CORE) program (Center No. 4/11).

References

Yasin Abbasi, Peter L Bartlett, Varun Kanade, Yevgeny Seldin, and Csaba Szepesvari. Online learning in markov decision processes with adversarially chosen transition probability distributions. In *Advances in Neural Information Processing Systems*, pages 2508–2516, 2013.

Noga Alon, Nicolò Cesa-Bianchi, Ofer Dekel, and Tomer Koren. Online learning with feedback graphs: Beyond bandits. In *Proceedings of The 28th Conference on Learning Theory*, pages 23–35, 2015.

Jean-Yves Audibert and Sébastien Bubeck. Minimax policies for adversarial and stochastic bandits. In *COLT*, pages 217–226, 2009.

P. Auer, R. Ortner, and C. Szepesvári. Improved rates for the stochastic continuum-armed bandit problem. *Proceedings of the 20th Annual Conference on Learning Theory*, pages 454–468, 2007.

Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.

Maria-Florina Balcan and Avrim Blum. Approximation algorithms and online mechanisms for item pricing. In *Proceedings of the 7th ACM Conference on Electronic Commerce*, pages 29–35. ACM, 2006.

Maria-Florina Balcan and Florin Constantin. Sequential item pricing for unlimited supply. In *International Workshop on Internet and Network Economics*, pages 50–62. Springer Berlin Heidelberg, 2010.

Maria-Florina Balcan, Avrim Blum, and Yishay Mansour. Item pricing for revenue maximization. In *Proceedings of the 9th ACM conference on Electronic commerce*, pages 50–59. ACM, 2008.

Nikhil Bansal, Ning Chen, Neva Cherniavsky, Atri Rurda, Baruch Schieber, and Maxim Sviridenko. Dynamic pricing for impatient bidders. *ACM Transactions on Algorithms (TALG)*, 6(2):35, 2010.

Yair Bartal. Probabilistic approximations of metric spaces and its algorithmic applications. In *37th Annual Symposium on Foundations of Computer Science, FOCS '96, Burlington, Vermont, USA, 14-16 October, 1996*, pages 184–193, 1996.

Omar Besbes and Assaf Zeevi. Dynamic pricing without knowing the demand function: Risk bounds and near-optimal algorithms. *Operations Research*, 57(6):1407–1420, 2009.

S. Bubeck, R. Munos, G. Stoltz, and C. Szepesvári. $\mathcal{X}$-armed bandits. *Journal of Machine Learning Research*, 12:1587–1627, 2011.

Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012.

Sébastien Bubeck, Ronen Eldan, and Yin Tat Lee. Kernel-based methods for bandit convex optimization. *arXiv preprint arXiv:1607.03084*, 2016.

Nicolo Cesa-Bianchi, Yishay Mansour, and Gilles Stoltz. Improved second-order bounds for prediction with expert advice. *Machine Learning*, 66(2-3):321–352, 2007.

E.W. Cope. Regret and convergence bounds for a class of continuum-armed bandit problems. *IEEE Transactions on Automatic Control*, 54(6):1243–1253, 2009.

Varsha Dani, Sham M Kakade, and Thomas P Hayes. The price of bandit information for online optimization. In *Advances in Neural Information Processing Systems*, pages 345–352, 2007.

Ofer Dekel, Jian Ding, Tomer Koren, and Yuval Peres. Bandits with switching costs: $T^{2/3}$ regret. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 459–467, 2014a.

Ofer Dekel, Elad Hazan, and Tomer Koren. The blinded bandit: Learning with adaptive feedback. In *Advances in Neural Information Processing Systems*, pages 1610–1618, 2014b.

Michal Feldman, Tomer Koren, Roi Livni, Yishay Mansour, and Aviv Zohar. Online pricing with strategic and patient buyers. In *Annual Conference on Neural Information Processing Systems*, 2016.

Sudipto Guha and Kamesh Munagala. Multi-armed bandits with metric switching costs. In *International Colloquium on Automata, Languages, and Programming*, pages 496–507. Springer, 2009.

Robert Kleinberg and Aleksandrs Slivkins. Sharp dichotomies for regret minimization in metric spaces. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 827–846. Society for Industrial and Applied Mathematics, 2010.

Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Multi-armed bandits in metric spaces. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 681–690. ACM, 2008.

Robert D. Kleinberg. Nearly tight bounds for the continuum-armed bandit problem. In *Advances in Neural Information Processing Systems*, pages 697–704, 2004.

Robert D. Kleinberg and Frank Thomson Leighton. The value of knowing a demand curve: Bounds on regret for online posted-price auctions. In *44th Symposium on Foundations of Computer Science FOCS*, pages 594–605, 2003.

Aleksandrs Slivkins. Multi-armed bandits on implicit metric spaces. In *Advances in Neural Information Processing Systems*, pages 1602–1610, 2011.

Aleksandrs Slivkins, Filip Radlinski, and Sreenivas Gollapudi. Ranked bandits in metric spaces: learning diverse rankings over large document collections. *Journal of Machine Learning Research*, 14(Feb):399–436, 2013.

J.Y. Yu and S. Mannor. Unimodal bandits. In *Proceedings of the 28th International Conference on Machine Learning*, 2011.