Multi-scale Online Learning: Theory and Applications to Online Auctions and Pricing

Sébastien Bubeck SEBUBECK@MICROSOFT.COM
Microsoft Research,
1 Microsoft Way,
Redmond, WA 98052, USA.

Nikhil Devanur NIKDEV@MICROSOFT.COM
Microsoft Research,
1 Microsoft Way,
Redmond, WA 98052, USA.

Zhiyi Huang ZHIYI@CS.HKU.HK
Department of Computer Science,
The University of Hong Kong,
Pokfulam, Hong Kong.

Rad Niazadeh RAD@CS.STANFORD.EDU
Department of Computer Science,
Stanford University,
Stanford, CA 94305, USA.

Editor: Csaba Szepesvari

Abstract

We consider revenue maximization in online auction/pricing problems. A seller sells an identical item in each period to a new buyer, or a new set of buyers. For the online pricing problem, both when the arriving buyer bids or only responds to the posted price, we design algorithms whose regret bounds *scale with the best fixed price in-hindsight*, rather than the range of the values. Under the bidding model, we further show our algorithms achieve a revenue convergence rate that matches the offline sample complexity of the single-item single-buyer auction. We also show regret bounds that are *scale free*, and match the offline sample complexity, when comparing to a benchmark that requires a lower bound on the market share. We further expand our results beyond pricing to multi-buyer auctions, and obtain online learning algorithms for auctions, with convergence rates matching the known sample complexity upper bound of online single-item multi-buyer auctions.

These results are obtained by generalizing the classical learning from experts and multi-armed bandit problems to their *multi-scale* versions. In this version, the reward of each action is in a different range, and the regret with respect to a given action scales with its own range, rather than the maximum range. We obtain almost optimal multi-scale regret bounds by introducing a new Online Mirror Descent (OMD) algorithm whose mirror map is the multi-scale version of the negative entropy function. We further generalize to the bandit setting by introducing the stochastic variant of this OMD algorithm.

Keywords: online learning, multi-scale learning, auction theory, bandit information, sample complexity. †

†. Following the theoretical computer science convention, we used **alphabetical author ordering.**

1. Introduction

Consider the following revenue maximization problem in a repeated setting, called the *online posted pricing* problem. In each period, the seller has a single item to sell, and a new prospective buyer. The seller offers to sell the item to the buyer at a given price; the buyer buys the item if and only if the price is below his private valuation for the item. The private valuation of the buyer itself is never revealed to the seller. How should a monopolistic seller iteratively set the prices if he wishes to maximize his revenue? What if he also cares about the market share, i.e. the fraction of time periods at which the item is sold?

Estimating price sensitivities and demand models in order to optimize revenue and market share is one of the bedrock of revenue management. The emergence of online marketplaces has enabled sellers to costlessly change prices, as well as collect huge amounts of data. This has renewed the interest in understanding best practices for data driven pricing. The extreme case of this when the price is updated for each buyer is the online pricing problem described above; one can always use this for less frequent price updates. Moreover this problem is intimately related to the classical experimentation and estimation procedures.

This problem has been studied from an *online learning* perspective, as a variant of the *multi-armed bandit* problem. In this variant, there is an arm for each possible price (presumably after an appropriate discretization). The revenue of each arm p is either p or zero, depending on whether the arriving value is at least equal to the price p or smaller than the price p, respectively. The total revenue of a pricing algorithm is then compared to the total revenue of the best fixed posted price in hindsight. The difference between the two, called the *regret*, is then bounded from above. No assumption is made on the distribution of values; the regret bounds are required to hold for the *worst case* sequence of values. Blum et al. (2004) assume that the buyer valuations are in $[1, h]$, and show the following multiplicative plus additive bound on the regret: for any $\epsilon \in (0, 1)$, the regret is at most ϵ times the revenue of the optimal price, plus $O(\epsilon^{-2} h \log h \log\log h)$. Blum and Hartline (2005) show that the additive factor can be made to be $O(\epsilon^{-3} h \log\log h)$, trading off a $\log h$ factor for an extra ϵ^{-1} factor.

An undesirable aspect of these bounds is that they scale *linearly with* h; this is particularly problematic when h is an estimate and we might set it to be a generous upper bound on the range of prices we wish to consider. A typical use case is when the same algorithm is used for many different products, with widely varying price ranges. We may not be able to manually tune the range for each product separately.

One might wonder if this dependence on h is unavoidable, as it seems to be reflected by the existing lower bounds for this problem in the literature (lower bounds are discussed later in the introduction with more details). Interestingly, in all of these lower-bound instances the best fixed price is equal to h itself; Therefore, it is not clear whether this dependency on h is required for instances where h is only a pessimistic upper-bound on the best fixed price. We now ask the following question:

> **Question**: *do online learning algorithms exist for the online posted pricing problem, such that their regrets are proportional to the best fixed price instead of the highest value?*

Standard off-the-shelf bounds allow regret to depend on the loss of the best arm instead of the worst case loss. However, even such bounds still depend linearly on the maximum range of all the losses, and thus they would not allow to replace h by the best fixed price.

Fortunately, in the online pricing problem the reward function of the arms is well structured. In particular, as a neat observation, the reward of the arm p is upper-bounded by p (and not only the maximum value). Can we use this structure in our favor to improve the standard regret bounds? We answer this question in the affirmative by the means of reducing the problem to a pure learning problem termed as *mutli-scale online learning.*

1.1 Multi-scale online learning

The main technical ingredients in our results are variants of the classical problems of learning from expert advice and multi-armed bandit. We introduce the multi-scale versions of these problems, where each action has its reward bounded in a different range. Here, we seek to design online learning algorithms that guarantee *multi-scale regret* bounds, i.e. their regrets with respect to each certain action scales with the range of that particular action, instead of the maximum possible range. These guarantees are in contrast with the regret bounds of the standard versions, which scale with the maximum range.

> **Main result (informal):** *we give algorithms for the full information and bandit information versions of the multi-scale online learning problem with information theoretically optimal (or almost optimal) multi-scale regret guarantees.*

While we use these bounds mostly for designing online auctions and pricing mechanisms, we expect such bounds to be of independent interest.

The main idea behind our algorithms is to use a tailored variant of *online (stochastic) mirror descent (OSMD)* (Bubeck, 2011). In this tailored version, the algorithm uses a *weighted negative entropy* as the Legendre function (also known as the *mirror map*), where the weight of each term i (corresponding to arm i) is actually equal to the range of that arm. More formally, assuming the range of arm i is equal to c_i, our mirror descent algorithms (Algorithm 1 for full information, and Algorithm 3 for the bandit information) use the following mirror map:

$$F(x) = \sum_{\text{arms } i} c_i \cdot x_i \ln(x_i)$$

Intuitively speaking, these algorithms take into account different ranges for different arms by first normalizing the reward of each arm by its range (i.e. divide the reward of arm i by its corresponding range c_i), and then projecting the updated weights by performing a *smooth multi-scale projection* into the simplex. This projection is an instance of the more general Bregman projection (Bubeck, 2011) for the special case of weighted negative entropy as the mirror map. The mirror descent framework then gives regret bounds in terms of a "local norm" as well as an "initial divergence", which we then bound differently for each version of the problem. In the technical sections we highlight how the subtle variations arise as a result of different techniques used to bound these two terms.

While our algorithms have the style of the multiplicative weights update (up to a normalization of the rewards), the smooth projection step at each iteration makes them drastically different. To shed some insight on this projection step, which plays an important role in our analysis, consider a very special case of the problem where the reward of each arm i is deterministically equal to c_i. The multiplicative weights algorithm picks arm i with a probability proportional to $\exp(c_i)$. However, as it is clear from the description of Algorithm 1, our algorithm uniformly scales the weight of each arm first. Then, in the projection step the weight of each arm i is multiplied by $\exp(-\frac{\lambda^*}{c_i})$ for some parameter λ^*. Hence, arm i will be sampled with a probability proportional to $\exp(-\frac{\lambda^*}{c_i})$ (which is a smooth approximation to $i^* = \text{argmax}\ c_i$, but in a different way compared to the vanilla multiplicative weights).

The multi-scale versions exhibit subtle variations that do not appear in the standard versions. First of all, our applications to auctions and pricing have non-negative rewards, and this actually makes a difference. For both the expert and the bandit versions, the minimax regret bounds for non-negative rewards are *provably better* than those when rewards could be negative. Further, for the bandit version, we can prove a better bound if we only require the bound to hold with respect to the *best* action, rather than *all* actions (for non-negative rewards). The various regret bounds and comparison to standard bounds are summarized in Tables 1.

	Standard regret bound $O(\cdot)$	Multi-scale bound (this paper)	
		Upper bound $O(\cdot)$	Lower bound $\Omega(\cdot)$
Experts/non-negative	$c_{\max}\sqrt{T\log(k)}$ *	$c_i\sqrt{T\log(kT)}$	$c_i\sqrt{T\log(k)}$
Bandits/non-negative	$c_{\max}\sqrt{Tk}$ †	$c_i T^{\frac{2}{3}}(k\log(kT))^{\frac{1}{3}}$	$c_i\sqrt{TK}$
		$c_{i^*}\sqrt{Tk\log(k)}$, i^* is the best action	-
Experts/symmetric	$c_{\max}\sqrt{T\log(k)}$ *	$c_i\sqrt{T\log(k\cdot\frac{c_{\max}}{c_{\min}})}$	$c_i\sqrt{T\log(k)}$
Bandits/symmetric	$c_{\max}\sqrt{Tk}$ †	$c_i\sqrt{Tk\cdot\frac{c_{\max}}{c_{\min}}\log(kT\cdot\frac{c_{\max}}{c_{\min}})}$	$c_i\sqrt{Tk\cdot\frac{c_{\max}}{c_{\min}}}$

* Freund and Schapire (1995); † Audibert and Bubeck (2009).

Table 1: Pure-additive regret bounds for non-negative rewards, i.e. when reward of any action i at any time is in $[0, c_i]$, and symmetric range rewards, i.e. when reward of any action i at any time is in $[-c_i, c_i]$ (suppose T is the time horizon, A is the action set, and k is the number of actions).

1.2 The implications for online auctions and pricing

As a direct application of our multi-scale online learning framework, somewhat surprisingly,

> **Second contribution:** *we show that we can get regret proportional to the best fixed price instead of the highest value for the online posted pricing problem.*

(i.e., we can replace h by the best fixed price, which is used in the definition of the benchmark). In particular, we show that the additive bound can be made to be $O(\epsilon^{-2}p^*\log h)$, where p^* is the best fixed price in hindsight. This allows us to use a very generous estimate for h and let the algorithm adapt to the actual range of prices; we only lose a $\log h$ factor.

The algorithm balances exploration probabilities of different prices carefully and automatically zooms in on the relevant price range. This does not violate known lower bounds, since in those instances p^* is close to h.

Bar-Yossef et al. (2002), Blum et al. (2004), and Blum and Hartline (2005) also consider the "full information" version of the problem, or what we call the *online (single buyer) auction* problem, where the valuations of the buyers are revealed to the algorithm after the buyer has made a decision. Such information may be available in a context where the buyers have to bid for the items, and are awarded the item if their bid is above a hidden price. In this case, the additive term can be improved to $O(\epsilon^{-1}h\log(\epsilon^{-1}))$, which is tight. Once again, by a reduction to multi-scale online learning, *we show that h can be replaced with p^**; in particular, we show that the additive term can be made to be $O(\epsilon^{-1}p^*\log(h\epsilon^{-1}))$.

1.3 Purely multiplicative bounds and sample complexity

The regret bounds mentioned above can be turned into a purely multiplicative factor in the following way: for any $\epsilon > 0$, the algorithm is guaranteed to get a $1 - O(\epsilon)$ fraction of the best fixed price revenue, provided the number of periods $T \geq E/\epsilon$, where E is the additive term in the regret bounds above. This follows from the observation that a revenue of T is a lower bound on the best fixed price revenue. Define the number of periods required to get a $1-\epsilon$ multiplicative approximation (as a function of ϵ) to be the *convergence rate* of the algorithm.

A $1-\epsilon$ multiplicative factor is also the target in the recent line of work, on the *sample complexity* of auctions, started by Balcan et al. (2008); Elkind (2007); Dhangwatnotai et al. (2014); Cole and Roughgarden (2014). (We give a more comprehensive discussion of this line of work in Section 1.4.) Here, *i.i.d.* samples of the valuations are given from a *fixed but unknown distribution*, and the goal is to find a price such that its revenue with respect to the hidden distribution is a $1-\epsilon$ fraction of the optimum revenue for this distribution. The sample complexity is the minimum number of samples needed to guarantee this (as a function of ϵ).

The sample complexity and the convergence rate (for the full information setting) are closely related to each other. The sample complexity is always smaller than the convergence rate: the problem is easier because of the following.

1. The valuations are i.i.d. in the case of sample complexity, whereas they can be arbitrary (worst case) in the case of convergence rate.

2. Sample complexity corresponds to an offline problem: you get all the samples at once. Convergence rate corresponds to an online problem: you need to decide what to do on a given valuation without knowing what valuations arrive in the future.

This is formalized in terms of an *online to offline reduction* [folklore] which shows that a convergence rate upper bound can be automatically translated to a sample complexity upper bound. This lets us convert sample complexity lower bounds into lower bounds on the convergence rate, and in turn into lower bounds on the additive error E in an additive plus multiplicative regret bound. For example, the additive error for the online auction problem (and hence also for the posted pricing problem[1]) cannot be $o(h\epsilon^{-1})$ (Huang et al., 2015b). Moreover, it is insightful to compare convergence rates we show with *the best known sample complexity upper bound; proving better convergence rates would mean improving these bounds as well.*

A natural target convergence rate for a problem is therefore the corresponding sample complexity, but achieving this is not always trivial. In particular, we consider an interesting version of the sample complexity bound for auctions, for which no analogous convergence rate bound is known in the literature. This version takes into account *both revenue and market share*, and gets sample complexity bounds that are *scale free*; there is no dependence on h, which means it works for unbounded valuations! For any $\delta \in (0, 1)$, the best fixed price benchmark is relaxed to ignore those prices whose market share (which is equivalent to the probability of sale) is below a δ fraction; as δ increases the benchmark is lower. This is a meaningful benchmark since in many cases revenue is not the only goal, even if you are a monopolist. A more reasonable goal is to maximize revenue subject to the constraint that the market share is above a certain threshold. What is more, this gives a sample complexity of $O(\epsilon^{-2}\delta^{-1}\log(\delta^{-1}\epsilon^{-1}))$ (Huang et al., 2015b). In fact δ can be set to h^{-1} without loss of generality, when the values are in $[1, h]$,[2] and the above bound then matches the sample complexity with respect to the best fixed price revenue. In addition, this bound gives a precise interpolation: as the target market share δ increase, the number of samples needed decreases almost linearly.

> **Third contribution:** *we show a convergence rate that almost matches the above sample complexity, for the full information setting.*

We have a mild dependence on h; the rate is proportional to $\log\log h$. Further, we also show a near optimal convergence rate for the online posted pricing problem.[3]

Multiple buyers: All of our results in the full information (online auction) setting extend to the multiple buyer model. In this model, in each time period, a new set of n buyers competes for a single item. The seller runs a truthful auction that determines the winning buyer and his payment. The benchmark here is the set of all "Myerson-type" mechanisms. These are mechanisms that are optimal when each period has n buyers of potentially different types, and the value of each buyer is drawn independently from a type dependent distribution. In fact, our convergence rates also imply new sample complexity bounds for these problems (except that they are not computationally efficient).

The various bounds and comparisons to previous work are summarized in Tables 2 & 3.

1.4 Other related work

The online pricing problem, also called *dynamic pricing*, is a much studied topic, across disciplines such as operations research and management science (Talluri and Van Ryzin,

1. We conjecture that the lower bound for the posted pricing problem should be worse by a factor of ϵ^{-1}, since one needs to explore about ϵ^{-1} different prices.
2. When the values are in $[1, h]$, we can guarantee a revenue of T by posting a price of 1, and to beat this, any other price (and in particular a price of h) would have to sell at least T/h times.
3. Unfortunately, we cannot yet guarantee that our online algorithm itself gets a market share of δ, although we strongly believe that it does. Showing such bounds on the market share of the algorithm is an important avenue for future research.

	Lower bound	Upper bound		
		Best known (Sample complexity)	Best known (Convergence rate)	This paper (Thm. 16)
Online single buyer auction	$\Omega\big(\frac{h}{\epsilon^2}\big)$ *	$\tilde{O}\big(\frac{h}{\epsilon^2}\big)$ †	$\tilde{O}\big(\frac{h}{\epsilon^2}\big)$ †	$\tilde{O}\big(\frac{p^*}{\epsilon^2}\big)$
Online posted pricing	$\Omega\big(\max\{\frac{h}{\epsilon^2}, \frac{1}{\epsilon^3}\}\big)$ *§	-	$\tilde{O}\big(\frac{h}{\epsilon^3}\big)$ †	$\tilde{O}\big(\frac{p^*}{\epsilon^3}\big)$
Online multi buyer auction	$\Omega\big(\frac{h}{\epsilon^2}\big)$ *	$O\big(\frac{nh}{\epsilon^3}\big)$ ‡	-	$\tilde{O}\big(\frac{nh}{\epsilon^3}\big)$

* Huang et al. (2015b); † Blum et al. (2004); ‡ Devanur et al. (2016); Gonczarowski and Nisan (2017); Elkind (2007); § Kleinberg and Leighton (2003).

Table 2: Number of rounds/samples needed to get a $1-\epsilon$ approximation to the best offline price/mechanism. Sample complexity is for the offline case with i.i.d. samples from an unknown distribution. Convergence rate is for the online case with a worst case sequence. Sample complexity is always no larger than the convergence rate. Lower bounds hold for sample complexity too, except for the online posted pricing problem for which there is no sample complexity version. The additive plus multiplicative regret bounds are converted to convergence rates by dividing the additive error by ϵ. In the last row, n is the number of buyers. In the last column, p^* denotes the optimal price.

	Lower bound (Sample complexity)	Upper bound	
		Best known (Sample complexity)	This paper (Thm. 17)
Online single buyer auction	$\Omega\big(\frac{1}{\epsilon^2\delta}\big)$ *	$\tilde{O}\big(\frac{1}{\epsilon^2\delta}\big)$ *	$\tilde{O}\big(\frac{1}{\epsilon^2\delta}\big)$
Online posted pricing	$\Omega\big(\max\{\frac{1}{\epsilon^2\delta}, \frac{1}{\epsilon^3}\}\big)$ *†	-	$\tilde{O}\big(\frac{1}{\epsilon^4\delta}\big)$
Online multi buyer auction	$\Omega\big(\frac{1}{\epsilon^2\delta}\big)$ *	-	$\tilde{O}\big(\frac{n}{\epsilon^3\delta}\big)$

* Huang et al. (2015b); † Kleinberg and Leighton (2003).

Table 3: Sample complexity & convergence rate w.r.t. the opt mechanism/price with market share $\geq \delta$.

2006), economics (Segal, 2003), marketing, and of course computer science. The multi-armed bandit approach to pricing is particularly popular. See den Boer (2015) for a recent survey on various approaches to the problem.

Kleinberg and Leighton (2003) consider the online pricing problem, under the assumption that the values are in $[0, 1]$, and considered purely additive factors. They showed that the minimax additive regret is $\tilde{\Theta}(T^{2/3})$, where T is the number of periods. This is similar in spirit to regret bounds that scale with h, since one has to normalize the values so that they are in $[0, 1]$. The finer distinction about the magnitude of the best fixed price is absent in this work. Recently, Syrgkanis (2017) also consider the online auction problem, with an emphasis on a notion of "oracle based" computational efficiency. They assume the values are all in $[0, 1]$ and do not consider the scaling issue that we do; this makes their contribution orthogonal to ours.

Starting with Dhangwatnotai et al. (2014), there has been a spate of recent results analyzing the sample complexity of pricing and auction problems. Cole and Roughgarden (2014) and Devanur et al. (2016) consider multiple buyer auctions with regular distributions (with unbounded valuations) and give sample complexity bounds that are polynomial in n and ϵ^{-1}, where n is the number of buyers. Morgenstern and Roughgarden (2015) consider arbitrary distributions with values bounded by h, and gave bounds that are polynomial in n, h, and ϵ^{-1}. Roughgarden and Schrijvers (2016); Huang et al. (2015b) give further improvements on the single- and multi-buyer versions respectively; Tables 2 and 3 give a comparison of these results with our bounds, for the problems we consider. The dynamic pricing problem has also been studied when there are a given number of copies of the item to sell (limited supply) (Agrawal and Devanur, 2014; Babaioff et al., 2015; Badanidiyuru et al., 2013; Besbes and Zeevi, 2009). There are also variants where the seller interacts with the same buyer repeatedly, and the buyer can strategize to influence his utility in the future periods (Amin et al., 2013).

Foster et al. (2017) also consider the multi-scale online learning problem motivated by a model selection problem. They consider additive bounds, for the symmetric case, for full information, but not bandit feedback. Their regret bounds are not comparable to ours in general; our bounds are better for the pricing/auction applications we consider, and their bounds are better for their application.

Organization We start in Section 2 by showing regret upper bounds for the multi-scale experts problem with non-negative rewards (Theorem 1). The corresponding upper bounds for the bandit version are in section 3 (Theorem 12). In Section 4 we show how the multi-scale regret bounds (Theorems 1 and 12) imply the corresponding bounds for the auction/pricing problems (Theorems 16 and 17). Finally, the regret (upper and lower) bounds for the symmetric range are discussed in Section 5 (Theorems 18, 20, 21, and 23).

2. Full Information Multi-scale Online Learning

We consider a variety of online algorithmic problems that are all parts of the *multiscale online learning* framework. We start by defining this framework, in which different actions have different ranges. We exploit this structure and express our results in terms of *action-specific* regret bounds for this general problem. To obtain these results, we use a variant of online mirror descent and propose a multiplicative-weight update style learning algorithm for our problem, termed as *Multi-Scale Multiplicative-Weight (MSMW)* algorithm.

Next, we investigate the single buyer auction problem (or equivalently the full-information single buyer dynamic pricing problem) as a canonical application, and show how to get multiplicative cum additive approximations here by the help of the multi-scale online learning framework. To show the tightness of our bounds, we compare the convergence rate of our dynamic pricing with the sample complexity of a closely related offline problem, i.e. the near optimal Bayesian revenue maximization from samples (Cole and Roughgarden, 2014).

2.1 The framework

Our full-information multi-scale online learning framework is basically the classical learning from expert advice problem. The main difference is that the *range* of rewards of different experts could be different. More formally, suppose there is a set of actions A.[4] The online problem proceeds in T rounds, where in each round $t \in [T]$:[5]

- The adversary picks a reward function $\mathbf{g}(t)$, where $g_i(t)$ is the reward of action i.
- The algorithm picks an action $i_t \in A$ simultaneously.
- Then the algorithm gets the reward $g_{i_t}(t)$ and observes the entire reward function $\mathbf{g}(t)$.

The total reward of the algorithm is denoted by

$$G_{\text{ALG}} := \textstyle\sum_{t=1}^{T} g_{i_t}(t).$$

The standard "best fixed action" benchmark is

$$G_{\text{MAX}} := \max_{i\in A} \textstyle\sum_{t=1}^{T} g_i(t).$$

We further assume that the action set is finite. Without loss of generality, if the action set is of size k, we identify $A = [k]$. The reward $\mathbf{g}(t)$ is such that for all $i \in A$, $g_i(t) \in [0, c_i]$, where $c_i \in \mathbb{R}_+$ is the *range* of action i.

2.2 Multi-scale regret bounds

We prove action-specific regret bounds, which we call also *multi-scale regret guarantees.* Towards this end, we define the following quantities.

$$G_i \;:=\; \textstyle\sum_{t\in[T]} g_i(t)\ , \tag{1}$$

$$\text{REGRET}_i \;:=\; G_i - G_{\text{ALG}}\ . \tag{2}$$

The regret bound w.r.t. action i, i.e., an upper bound on $\mathbb{E}\left[\text{REGRET}_i\right]$, depends on the range c_i, as well as any *prior* distribution $\boldsymbol{\pi}$ over the action set A; this way, we can handle countably many actions. Let $c_{\min} = \inf_{i\in A} c_i$ and $c_{\max} = \sup_{i\in A} c_i$ (if applicable) be the minimum and the maximum range. We first state a version of the regret bound which is parameterized by $\epsilon > 0$; such bounds are stronger than $\sqrt{T}$ type bounds which are more standard.

Theorem 1 (Main Result) *There exists an algorithm for the full-information multi-scale online learning problem that takes as input any distribution* $\boldsymbol{\pi}$ *over* A*, the ranges* c_i, $\forall\, i \in A$ *and a parameter* $0 < \epsilon \leq 1$*, and satisfies:*

$$\forall i \in A: \quad \mathbb{E}\left[\text{REGRET}_i\right] \leq \epsilon \cdot G_i + O\left(\frac{1}{\epsilon} \log\left(\frac{1}{\epsilon \pi_i}\right) \cdot c_i\right) \tag{3}$$

4. We use the terms experts, arms and actions interchangeably in this paper.
5. We use the notation $[n] := \{1, 2, \ldots, n\}$, for any $n \in \mathbb{N}$.

Compare this to what you get by using the standard analysis for the experts problem (Arora et al., 2012), where the second term in the regret bound is $O\big(\frac{1}{\epsilon}\log(k)\cdot c_{\max}\big)$. Choosing $\boldsymbol{\pi}$ to be the uniform distribution in the above theorem gives $O\big(\frac{1}{\epsilon}\log\left(\frac{k}{\epsilon}\right)\cdot c_i\big)$. Also, one can compare the pure-additive version of this bound with the classic pure-additive regret bound $O\big(c_{\max}\cdot\sqrt{T\log(k)}\big)$ for the experts problem by setting $\epsilon=\sqrt{\frac{\log(kT)}{T}}$ (Corollary 2).

Corollary 2 *There exists an algorithm for the full-information multi-scale online learning problem that takes as input the ranges c_i, $\forall\ i\in A$, and satisfies:*

$$\forall i\in A: \quad \mathbb{E}\left[\textsc{regret}_i\right]\leq O\left(c_i\cdot\sqrt{T\log(kT)}\right) \tag{4}$$

Remark 3 *We should assert that in a multi-scale regret guarantee, we provide a separate regret bound for each action, where the bound on the regret of action i only scales linearly with c_i. This type of guarantee should "not" be mistaken as a bound on the worst action.*

Here is the map of the rest of this section. In Section 2.3 we propose an algorithm that exploits the reward structure, and later in Section 2.4 we show how this algorithm is an online mirror descent with weighted negative entropy as its mirror map. For reward-only instances, we prove the regret bound in Section 2.5. We finally turn our attention to the single buyer online auction problem in Section 2.6.

2.3 Multi-Scale Multiplicative-Weight (MSMW) algorithm

We achieve our regret bound in Theorem 1 by using the MSMW algorithm (Algorithm 1). The main idea behind this algorithm is to take into account different ranges for different experts, and therefore:

1. We normalize the reward of each expert accordingly, i.e. divide the reward of expert i by its corresponding range c_i;

2. We project the updated weights by performing a *smooth multi-scale projection* into the simplex: the algorithm finds a λ^* such that multiplying the current weight of each expert i by $\exp\left(-\frac{\lambda^*}{c_i}\right)$ makes a probability distribution over the experts. It then uses this resulting probability distribution for sampling the next expert.

Algorithm 1 MSMW

input initial distribution $\boldsymbol{\mu}$ over A, learning rate $0<\eta\leq 1$.
initialize $\mathbf{p}(1)$ such that $p_i(1)=\mu_i$ for all $i\in A$.
for $t=1,\ldots,T$ **do**
  Randomly pick an action drawn from $\mathbf{p}(t)$, and observe $\mathbf{g}(t)$.
  $\forall i\in A:\quad w_i(t+1)\leftarrow p_i(t)\cdot\exp(\eta\cdot\frac{g_i(t)}{c_i})$.
  Find λ^* (e.g., binary search) s.t. $\sum_{i\in A}w_i(t+1)\cdot\exp(-\frac{\lambda^*}{c_i})=1$.
  $\forall i\in A:\quad p_i(t+1)\leftarrow w_i(t+1)\cdot\exp(-\frac{\lambda^*}{c_i})$.
end for

2.4 Equivalence to online mirror descent with weighted negative entropy

While it is possible to analyze the regret of the MSMW algorithm (Algorithm 1) by using first principles, we take a different approach (the elementary analysis can still be found in the appendix, Section A.2). We show how this algorithm is indeed an instance of the Online Mirror Descent (OMD) algorithm for a particular choice of the *Legendre function* (also known as *the mirror map*).

2.4.1 Preliminaries on online mirror descent.

Fix an open convex set $\mathcal{D}$ and its closure $\bar{\mathcal{D}}$, which in our case are $(0,+\infty)^A$ and $[0,+\infty)^A$ respectively, and a closed-convex action set $\mathcal{A} \subset \bar{\mathcal{D}}$, which in our case is Δ_A, i.e. the set of all probability distributions over experts in A. At the heart of an OMD algorithm there is a *Legendre* function $F : \bar{\mathcal{D}} \to \mathbb{R}$, i.e. a strictly convex function that admits continuous first order partial derivatives on $\mathcal{D}$ and $\lim_{x\to\bar{\mathcal{D}}\setminus\mathcal{D}} \|\nabla F(x)\| = +\infty$, where $\nabla F(.)$ denotes the gradient map of F. One can think of OMD as a member of *projected gradient descent* algorithms, where the *gradient update* happens in the *dual space* $\nabla F(\mathcal{D})$ rather than in primal $\mathcal{D}$, and the *projection* is defined by using the *Bregman divergence* associated with F rather than ℓ_2-distance (see Figure 2.4.1).

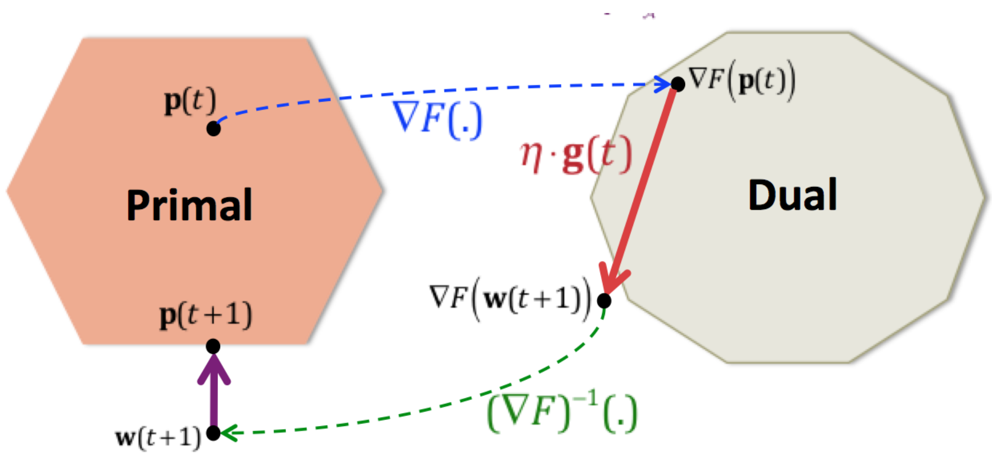

Figure 1: Online Mirror Descent (OMD): moving to the dual space by gradient map *(blue)*, gradient update in the dual space *(red)*, applying the inverse gradient map *(green)*, and finally projecting back to the simplex using Bregman projection *(purple)*.

Definition 4 (Bregman Divergence (Bubeck, 2011)) *Given a Legendre function F over Δ_A, the Bregman divergence associated with F, denoted as $D_F : \Delta_A \times \Delta_A \to \mathbb{R}$, is defined by*

$$D_F(x, y) = F(x) - F(y) - (x - y)^T \nabla F(y)$$

Definition 5 (Online Mirror Descent (Bubeck, 2011)) *Suppose F is a Legendre function. At every time $t \in [T]$, the online mirror descent algorithm with Legendre function F*

selects an expert drawn from distribution $\mathbf{p}(t)$*, and then updates* $\mathbf{w}(t)$ *and* $\mathbf{p}(t)$ *given rewards* $\mathbf{g}(t)$ *by:*

Gradient update:

$$\nabla F(\mathbf{w}(t+1)) = \nabla F(\mathbf{p}(t)) + \eta \cdot \mathbf{g}(t) \Rightarrow \mathbf{w}(t+1) = (\nabla F)^{-1}\left(\nabla F(\mathbf{p}(t)) + \eta \cdot \mathbf{g}(t)\right) \quad (5)$$

Bregman projection:

$$\mathbf{p}(t+1) = \underset{\mathbf{p}\in\Delta_A}{argmin}\left(D_F(\mathbf{p}, \mathbf{w}(t+1))\right) \quad (6)$$

where $\eta > 0$ *is called the learning rate of OMD.*

We use the following standard regret bound of OMD (Refer to Bubeck (2011) for a thorough discussion on OMD. For completeness, a proof is also provided in the appendix, Section A.3). Roughly speaking, this lemma upper-bounds the regret by the summation of two separate terms: "local norm" (the first term), which captures the total deviation between $\mathbf{p}(t)$ and $\mathbf{w}(t+1)$, and "initial divergence" (the second term), which captures how much the initial distribution is far from the target distribution.

Lemma 6 *For any learning rate parameter* $0 < \eta \leq 1$ *and any benchmark distribution* $\mathbf{q}$ *over* A*, the OMD algorithm with Legendre function* $F(.)$ *admits the following:*

$$\textstyle\sum_{t\in[T]} \mathbf{g}(t) \cdot \big(\mathbf{q} - \mathbf{p}(t)\big) \leq \frac{1}{\eta}\sum_{t\in[T]} D_F(\mathbf{p}(t), \mathbf{w}(t+1)) + \frac{1}{\eta} D_F(\mathbf{q}, \mathbf{p}(1)) \quad (7)$$

2.4.2 MSMW algorithm as an OMD

For our application, we focus on a particular choice of Legendre function that captures different learning rates proportional to c_i^{-1} for different experts, as we saw earlier in Algorithm 1. We start by defining the *weighted negative entropy* function.

Definition 7 *Given expert-ranges* $\{c_i\}_{i\in A}$*, the* weighted negative entropy *is defined by*

$$F(x) = \textstyle\sum_{i\in A} c_i \cdot x_i \ln(x_i) \quad (8)$$

Corollary 8 *It is straightforward to see* $F(x) = \sum_{i\in A} c_i \cdot x_i \ln(x_i)$ *is a non-negative Legendre function over* $\mathbb{R}_+^A$*. Moreover,* $\nabla F(x)_i = c_i(1 + \ln(x_i))$ *and* $D_F(x, y) = \sum_{i\in A} c_i \cdot (x_i \ln(\frac{x_i}{y_i}) - x_i + y_i)$*.*

We now have the following lemma that shows Algorithm 1 is indeed an OMD algorithm.

Lemma 9 *The MSMW algorithm, i.e. Algorithm 1, is equivalent to an OMD algorithm associated with the weighted negative entropy* $F(x) = \sum_{i\in A} c_i \cdot x_i \ln(x_i)$ *as its Legendre function.*

Proof Look at the gradient update step of OMD, as in Equation (5), with Legendre function $F(x) = \sum_{i\in A} c_i \cdot x_i \ln(x_i)$. By using Corollary 8 we have

$$\nabla F(\mathbf{w}(t+1)) = \nabla F(\mathbf{p}(t)) + \eta \cdot \mathbf{g}(t) \Rightarrow c_i(1 + \ln(w_i(t+1))) = c_i(1 + \ln(p_i(t))) + \eta \cdot g_i(t)\ ,$$

and therefore, $w_i(t+1) = p_i(t) \cdot \exp(\eta \cdot \frac{g_i(t)}{c_i})$. Moreover, for the Bregman projection step we have

$$\mathbf{p}(t+1) = \operatorname*{argmin}_{\mathbf{p}\in\Delta_A} \left(D_F(\mathbf{p}, \mathbf{w}(t+1))\right) = \operatorname*{argmin}_{\mathbf{p}\in\Delta_A} \left(\sum_{i\in A} c_i \cdot (p_i \ln(\frac{p_i}{w_i(t+1)}) - p_i + w_i(t+1))\right) \tag{9}$$

This is a convex minimization over a convex set. To find a closed form solution, we look at the Lagrangian dual function $\mathcal{L}(\mathbf{p}, \lambda) \triangleq \sum_{i\in A} c_i \cdot (p_i \ln(\frac{p_i}{w_i(t+1)}) - p_i + w_i(t+1)) + \lambda(\sum_{i\in A} p_i - 1)$ and the Karush-Kuhn-Tucker (KKT) conditions $\nabla\mathcal{L}(\mathbf{p}^*, \lambda^*) = \mathbf{0}$. We have

$$c_i \cdot \ln(\frac{p_i^*}{w_i(t+1)}) + \lambda^* = 0 \Rightarrow p_i^* = w_i(t+1) \cdot \exp(-\frac{\lambda^*}{c_i}) \tag{10}$$

As $\sum_{i\in A} p_i^* = 1$, λ^* should be unique number s.t. $\sum_{i\in A} w_i(t+1) \cdot \exp(-\frac{\lambda^*}{c_i}) = 1$, and then $p_i(t+1) = w_i(t+1) \cdot \exp(-\frac{\lambda^*}{c_i})$. So, Algorithm 1 is equivalent to OMD with weighted negative entropy as its Legendre function. ■

By combining Lemma 6, Corollary 8 and finally Lemma 9 we prove the following regret bound for the MSMW algorithm. We encourage the reader to also look at the appendix, Section A.2, for an extra proof using first principles.

Proposition 10 *For any initial distribution $\boldsymbol{\mu}$ over A, and any learning rate parameter $0 < \eta \leq 1$, and any benchmark distribution $\mathbf{q}$ over A, the MSMW algorithm satisfies that:*

$$\sum_{i\in A} q_i \cdot G_i - \mathbb{E}\left[G_{\text{ALG}}\right] \leq \eta \sum_{t\in[T]} \sum_{i\in A} p_i(t) \frac{(g_i(t))^2}{c_i} + \frac{1}{\eta} \cdot \sum_{i\in A} c_i \left(q_i \ln\left(\frac{q_i}{\mu_i}\right) - q_i + \mu_i\right) .$$

Proof [of Proposition 10] We have:

$$\sum_{i\in A} q_i \cdot G_i - \mathbb{E}\left[G_{\text{ALG}}\right] = \sum_{t\in[T]} \mathbf{q} \cdot \mathbf{g}(t) - \sum_{t\in[T]} \mathbf{p}(t) \cdot \mathbf{g}(t) = \sum_{t\in[T]} \mathbf{g}(t) \cdot \left(\mathbf{q} - \mathbf{p}(t)\right) \tag{11}$$

By applying the regret bound of OMD (Lemma 6) to upper-bound the RHS, we have

$$\sum_{i\in A} q_i \cdot G_i - \mathbb{E}\left[G_{\text{ALG}}\right] \leq \frac{1}{\eta} \sum_{t\in[T]} D_F(\mathbf{p}(t), \mathbf{w}(t+1)) + \frac{1}{\eta} D_F(\mathbf{q}, \mathbf{p}(1)) \tag{12}$$

To bound the first term in regret, a.k.a *local norm*, we have:

$$\begin{aligned} D_F(\mathbf{p}(t), \mathbf{w}(t+1)) &= \sum_{i\in A} c_i \cdot (p_i(t) \ln(\frac{p_i(t)}{w_i(t+1)}) - p_i(t) + w_i(t+1)) \\ &= \sum_{i\in A} c_i \cdot p_i(t)(-\eta \cdot \frac{g_i(t)}{c_i} - 1 + exp(\eta \cdot \frac{g_i(t)}{c_i})) \end{aligned} \tag{13}$$

Note that $\eta \cdot \frac{g_i(t)}{c_i} \in [-1,1]$ because $g_i(t) \in [-c_i, c_i]$ and $0 < \eta \leq 1$. By $\exp(x) - x - 1 \leq x^2$ for $-1 \leq x \leq 1$ and that $\eta g_i(t) \in [-c_i, c_i]$, the above is upper bounded by $\eta^2 \sum_{i \in A} p_i(t) \frac{(g_i(t))^2}{c_i}$. We can also rewrite the second term in regret. In fact, if we set $\mathbf{p}(1) = \boldsymbol{\mu}$, then

$$\frac{1}{\eta} \cdot D_F(\mathbf{q}, \mathbf{p}(1)) = \frac{1}{\eta} \cdot \sum_{i \in A} c_i \left(q_i \ln \left(\frac{q_i}{\mu_i}\right) - q_i + \mu_i \right)$$

By summing the upper-bounds $\eta^2 \sum_{i \in A} p_i(t) \frac{(g_i(t))^2}{c_i}$ on each term of local norm in (13) for $t \in [T]$ and putting all the pieces together, we get the desired bound. ■

2.5 Regret analysis for non-negative rewards

Theorem 1 *There exists an algorithm for the full-information multi-scale online learning problem that takes as input any distribution $\boldsymbol{\pi}$ over A, the ranges c_i, $\forall\, i \in A$ and a parameter $0 < \epsilon \leq 1$, and satisfies:*

$$\forall i \in A: \quad \mathbb{E}\left[\text{REGRET}_i\right] \leq \epsilon \cdot G_i + O\left(\frac{1}{\epsilon} \log\left(\frac{1}{\epsilon \pi_i}\right) \cdot c_i \right) \tag{14}$$

Proof [of Theorem 1] Suppose $i_{\min}$ is an action with the minimum c_i. Let $\boldsymbol{\mu} = (1-\eta) \cdot \mathbf{1}_{i_{\min}} + \eta \cdot \boldsymbol{\pi}$, and let $\mathbf{q} = (1-\eta) \cdot \mathbf{1}_i + \eta \cdot \boldsymbol{\pi}$ in Proposition 10. If $i \neq i_{\min}$, we get that (note that $\mu_j = q_j$ for any $j \neq i, i_{\min}$):

$$(1-\eta) \cdot G_i + \eta \cdot \sum_{j \in A} \pi_j \cdot G_j - \mathbb{E}\left[G_{\text{ALG}}\right] \leq \eta \cdot \mathbb{E}\left[G_{\text{ALG}}\right] + \frac{1}{\eta} \cdot c_i \cdot \left(q_i \ln\left(\frac{q_i}{\mu_i}\right) - q_i + \mu_i \right)$$
$$+ \frac{1}{\eta} \cdot c_{i_{\min}} \cdot \left(q_{i_{\min}} \ln\left(\frac{q_{i_{\min}}}{\mu_{i_{\min}}}\right) - q_{i_{\min}} + \mu_{i_{\min}} \right)$$

By $1 \geq q_i > \mu_i \geq \eta \pi_i$, the second term on the RHS is upper bounded as:

$$\frac{1}{\eta} \cdot c_i \cdot \left(q_i \ln\left(\frac{q_i}{\mu_i}\right) - q_i + \mu_i \right) \leq \frac{1}{\eta} \cdot c_i \cdot \ln\left(\frac{1}{\eta \pi_i}\right)$$

Similarly, by $1 \geq \mu_{i_{\min}} > q_{i_{\min}} \geq 0$, the third term on the RHS is upper bounded as

$$\frac{1}{\eta} \cdot c_{i_{\min}} \cdot \left(q_{i_{\min}} \ln\left(\frac{q_{i_{\min}}}{\mu_{i_{\min}}}\right) - q_{i_{\min}} + \mu_{i_{\min}} \right) \leq \frac{1}{\eta} \cdot c_{i_{\min}} \leq \frac{1}{\eta} \cdot c_i$$

Finally, note that $G_j \geq 0$ for all $j \in A$ in reward-only instances. So the LHS is lower bounded by

$$(1-\eta) \cdot G_i - \mathbb{E}\left[G_{\text{ALG}}\right] = (1-\eta) \cdot \text{REGRET}_i - \eta \cdot \mathbb{E}\left[G_{\text{ALG}}\right].$$

Putting all this together, we get that

$$\mathbb{E}\left[\text{REGRET}_i\right] \leq \frac{2\eta}{1-\eta} \cdot \mathbb{E}\left[G_{\text{ALG}}\right] + O\left(\frac{1}{\eta} \ln\left(\frac{1}{\eta \pi_i}\right) \cdot c_i \right) \leq 3\eta \cdot \mathbb{E}\left[G_{\text{ALG}}\right] + O\left(\frac{1}{\eta} \ln\left(\frac{1}{\eta \pi_i}\right) \cdot c_i \right).$$

The theorem then follows by choosing $\eta = \frac{\epsilon}{3}$ and rearranging terms. ■

2.6 A canonical application: online single buyer auction

The setup. The simple auction design problem that we consider is as follows. There is a seller with infinite identical copies of an item. Buyers arrive over time. At each round, the seller picks a *price* and the arriving buyer reports her *value*. If the value is no less than the price, the trade happens; money goes to the seller and the copy of the item goes to the arriving buyer. The goal is to maximize the revenue of the seller.

Formally, we look at this problem as an instance of the full information multi-scale online learning framework; The action set is $A = [1, h]$. [6] The reward function is such that at round t the adversary (i.e. the arriving buyer) picks a value $v(t) \in [1, h]$ and for any price $p \in A$ picked by the seller (i.e. the algorithm), the reward is $g_p(t) := p \cdot \mathbf{1}(v(t) \geq p)$. This is a full information setting, because the value $v(t)$ is revealed to the algorithm after each round t.

The additive/multiplicative approximation. In order to obtain a $(1-\epsilon)$-approximation of the optimal revenue, i.e. the revenue of the best fixed price p^* in hindsight, it suffices to consider prices of the form $(1+\epsilon)^j$ for $0 \leq j \leq \lfloor \log_{1+\epsilon} h \rfloor = O(\frac{\log h}{\epsilon})$. As a result, we reduce the online single buyer auction problem to the multi-scale online learning with full information and finite actions. The action set has $k = O(\frac{\log h}{\epsilon})$ actions whose ranges form a geometric sequence $(1+\epsilon)^j$, $0 \leq j < k$.

Recall the definition of G_{MAX} in Section 2.1, and let p^* be the best fixed price in hindsight, which is the price that achieves G_{MAX}. We now show how to get a multiplicative cum additive approximation for this problem with G_{MAX} as the benchmark, à la Blum et al. (2004); Blum and Hartline (2005). The main improvement over these results is that the additive term scales with the best price rather than h.

Theorem 11 *There is an algorithm for the online single buyer auction problem that takes as input a parameter $\epsilon > 0$, and satsify $G_{\text{ALG}} \geq (1-\epsilon)G_{\text{MAX}} - O(E)$, where:*

$$E = \frac{p^* \log(\log h/\epsilon)}{\epsilon} \ .$$

Also, even if h is not known up front, there is an (slightly modified) algorithm that achieves a similar approximation guarantee for online single buyer auction with:

$$E = \frac{p^* \log(p^*/\epsilon)}{\epsilon} \ .$$

Proof [of Theorem 11]
[Part 1: known h] Recall the above formulation of the problem as an online learning problem with full information. The proof then follows by Theorem 1, letting $\boldsymbol{\pi}$ to be the uniform distribution over the $k = O(\log h/\epsilon)$ actions, i.e., discretized prices.
[Part 2: unknown h] When h is not known up front, we consider a variant of our algorithm (Algorithm 2) that picks the next price in each round t from the set of *relevant prices* (denoted by $\mathcal{P}$), updates this set if necessary, and then updates the weights of prices in this set as in Algorithm 1. The main new idea here is to update the set of prices $\mathcal{P}$ so that it only includes prices that are at most the highest value we have seen so far (let the highest seen value be 1 at the beginning). Now, for the sake of analysis, consider a hypothetical

6. Here, we allow an infinite action set. Later, we show how to discretize to get around this issue.

algorithm (called $\texttt{ALG}^H$) that considers a countably infinite action space comprising all prices of the form $(1+\epsilon)^j$, for $j \geq 0$. We first show this hypothetical algorithm $\texttt{ALG}^H$ satisfies the required approximation guarantee in Theorem 11. We then show the expected revenue of Algorithm 2 is at least the expected revenue of $\texttt{ALG}^H$ (minus a constant that is negligible in our bound), and hence the final proof.

The proof of the regret bound of Theorem 1 works when we have countably many actions (although we cannot implement such algorithms directly). Now, consider simulating $\texttt{ALG}^H$ and let the prior distribution $\boldsymbol{\pi}$ be such that for any price $p = (1+\epsilon)^j$, $\pi_p = \epsilon(\epsilon+2)(1+\epsilon)^{-2(j+1)} = \frac{\epsilon(\epsilon+2)}{(1+\epsilon)^2} \cdot \frac{1}{p^2}$ (this choice will become more clear later in the proof; in short we need π_p to be proportional to $\frac{1}{p^2}$). The approximation guarantee in Theorem 11 then follows by Theorem 1. We now argue the followings:

- For any round t, unless the value in that round is a new highest value, Algorithm 2 gets weakly higher revenue than $\texttt{ALG}^H$. This is because the probability that Algorithm 2 plays any relevant price in $\mathcal{P}$ (that has a non-zero gain in this round) is weakly higher than that in $\texttt{ALG}^H$.

- For any price $p = (1+\epsilon)^j$, consider the first time a value at least p shows up. Algorithm 2 suffers a loss of at most $p \cdot \pi_p$ compared to $\texttt{ALG}^H$, due to $\texttt{ALG}^H$'s probability of playing p in that round, where π_p is the probability of playing p in the initial distribution. This is because the probability that $\texttt{ALG}^H$ plays p in this round is at most π_p as p has not got any positive gains before this round.

- Then, by choosing π_p to be inversely proportional to p^2, we can show that Algorithm 2 has an additive loss of $\sum_p \frac{\beta}{p} = \frac{\epsilon+2}{\epsilon+1} = O(1)$ compared to$\texttt{ALG}^H$, where $\beta = \left(\sum_p \frac{1}{p^2}\right)^{-1} = \frac{\epsilon(2+\epsilon)}{(1+\epsilon)^2}$ is the normalization constant of the initial distribution $\boldsymbol{\pi}$. This finishes the proof.

■

Algorithm 2 Online single buyer auction (for unknown h)

```
input learning rate 0 < η ≤ 1, price discretization parameter 0 < ϵ ≤ 1.
initialize the set of relevant prices P = {1}. Let α_1(1) = 1.
for t = 1, ..., T do
    Randomly pick a price in P drawn from α(t), and observe g(t).
    Update P to be all the prices (1+ϵ)^j that are at most the highest value until time t.
    ∀p ∈ P :  w_p(t+1) ← α_p(t) · exp(η · g_p(t)/p).
    Find λ* (e.g., binary search) s.t. Σ_{p∈P} w_p(t+1) · exp(−λ*/p) = 1.
    ∀p ∈ P :  α_p(t+1) ← w_p(t+1) · exp(−λ*/p).
end for
```

Bounds on the sample complexity of auctions for single buyer problem (Huang et al., 2015a) imply that the first bound in this theorem is tight up to log factors: the lower bound is $h\epsilon^{-1}$ in an instance where p^* is actually equal to h. Also, the best upper bound known is by Blum et al. (2004); Blum and Hartline (2005), which is

$$E = \frac{h \log(1/\epsilon)}{\epsilon} .$$

We conclude that Theorem 11 generalizes the known tight sample complexity upper-bound for the offline single buyer Bayesian revenue maximization to the online adversarial setting.

3. Multi-Scale Online Learning with Bandit Feedback

In this section, we look at the bandit feedback version of multi-scale online learning framework proposed in Section 2.1. Essentially, the only difference here is that after the algorithm picks an arm i_t at time t, it only observes the obtained reward, i.e. $g_{i_t}(t)$, and *does not* observe the entire reward function $\mathbf{g}(t)$.

Inspired by the online stochastic mirror descent algorithm (Bubeck, 2011) we introduce *Bandit-MSMW* algorithm. Our algorithm follows the standard bandit route of using unbiased estimators for the rewards in a full information strategy (in this case MSMW). We also mix the MSMW distribution with an extra uniform exploration, and use a tailored initial distribution to obtain the desired mutli-scale regret bounds.

3.1 Bandit multi-scale regret bounds

For the bandit version, we can get similar regret guarantees as in Section 2.2 for the full-information variant, but only for the *best* action. If we require the regret bound to hold for all actions, then we can only get a weaker bound, where the second term has ϵ^{-2} instead of ϵ^{-1}. The difference between the bounds for the bandit and the full information setting is essentially a factor of k, which is unavoidable.

Theorem 12 *There exists an algorithm for the online multi-scale problem with bandit feedback that takes as input the ranges c_i, $\forall\ i \in A$, and a parameter $0 < \epsilon \leq 1$, and satisfies,*

- *for $i^* = \arg\max_{i \in A} G_i$,*

$$\mathbb{E}\left[\text{REGRET}_{i^*}\right] \leq \epsilon \cdot G_{i^*} + O\left(\tfrac{1}{\epsilon} k \log\left(\tfrac{k}{\epsilon}\right) \cdot c_{i^*}\right). \tag{15}$$

- *for all $i \in A$,*

$$\mathbb{E}\left[\text{REGRET}_{i}\right] \leq \epsilon \cdot G_{i} + O\left(\tfrac{1}{\epsilon^2} k \log\left(\tfrac{k}{\epsilon}\right) \cdot c_{i}\right). \tag{16}$$

Also, one can compute the pure-additive versions of the bounds in Theorems 12 by setting $\epsilon = \sqrt{\frac{k\log(kT)}{T}}$ and $\epsilon = (\frac{k\log(kT)}{T})^{\frac{1}{3}}$ resepctively (Corollary 13), and compare with the pure-additive regret bound $O\big(c_{\max} \cdot \sqrt{Tk}\big)$ for the adversarial multi-armed bandit problem (Audibert and Bubeck, 2009; Auer et al., 1995).

Corollary 13 *There exist algorithms for the online multi-scale bandits problem that satisfies,*

- *For $i^* = \arg\max_{i \in A} G_i$,*

$$\mathbb{E}\left[\text{REGRET}_{i^*}\right] \leq O\left(c_{i^*} \cdot \sqrt{Tk\log(kT)}\right) \tag{17}$$

- *For all $i \in A$,*

$$\mathbb{E}\left[\text{REGRET}_i\right] \leq O\left(c_i \cdot T^{\frac{2}{3}} (k \log(kT))^{\frac{1}{3}}\right) \tag{18}$$

Here is a map of this section. In Section 3.2 we propose our bandit algorithm and prove its general regret guarantee for non-negative rewards. Then in Section 3.3 we show how to get a multi-scale style regret guarantee for the best arm c_{i^*}, and a weaker guarantee for all arms $\{c_i\}_{i \in A}$.

3.2 Bandit Multi-Scale Multiplicative Weight (Bandit-MSMW) algorithm

We present our Bandit algorithm (Algorithm 3) when the set of actions A is finite (with $|A| = k$). Let η be the learning rate and γ be the exploration probability. We show the following regret bound.

Algorithm 3 Bandit-MSMW

input exploration parameter $\gamma > 0$, learning rate $\eta > 0$.
initialize $\mathbf{p}(1) = (1-\gamma)\mathbf{1}_{i_{\min}} + \frac{\gamma}{k}\mathbf{1}$, where $i_{\min}$ is the arm with minimum range $c_{i_{\min}}$.
for $t = 1, \ldots, T$ **do**
  Let $\tilde{\mathbf{p}}(t) = (1-\gamma)\mathbf{p}(t) + \frac{\gamma}{k}\mathbf{1}$.
  Randomly pick an expert i_t drawn from $\tilde{\mathbf{p}}(t)$, and observe $g_{i_t}(t)$.
  Let $\tilde{\mathbf{g}}(t)$ be such that

$$\tilde{g}_i(t) = \begin{cases} \frac{g_i(t)}{\tilde{p}_i(t)} & \text{if } i = i_t; \\ 0 & \text{otherwise.} \end{cases}$$

  $\forall i \in A: \quad w_i(t+1) \leftarrow p_i(t) \cdot \exp(\frac{\eta}{c_i} \cdot \tilde{g}_i(t))$.
  Find λ^* (e.g., binary search) s.t. $\sum_{i \in A} w_i(t+1) \cdot \exp(-\frac{\lambda^*}{c_i}) = 1$.
  $\forall i \in A: \quad p_i(t+1) \leftarrow w_i(t+1) \cdot \exp(-\frac{\lambda^*}{c_i})$.
end for

Lemma 14 *For any exploration probability $0 < \gamma \leq \frac{1}{2}$ and any learning rate parameter $0 < \eta \leq \frac{\gamma}{k}$, the Bandit-MSMW algorithm achieves the following regret bound when the gains are non-negative :*

$$\forall i \in A : \mathbb{E}\left[\text{REGRET}_i\right] \leq O\left(\frac{1}{\eta} \log\left(\frac{k}{\gamma}\right) \cdot c_i + \eta \textstyle\sum_{j \in A} G_j + \gamma \cdot G_i\right)$$

Proof [of Lemma 14] We further define:

$$\begin{aligned} \widetilde{G}_{\text{ALG}} &\triangleq \textstyle\sum_{t \in [T]} g_{i_t}(t) = \sum_{t \in [T]} \tilde{\mathbf{p}}(t) \cdot \tilde{\mathbf{g}}(t) \ , \\ \widetilde{G}_j &\triangleq \textstyle\sum_{t \in [T]} \tilde{g}_j(t) \ . \end{aligned}$$

In expectation over the randomness of the algorithm, we have:

1. $\mathbb{E}\left[G_{\text{ALG}}\right] = \mathbb{E}\left[\widetilde{G}_{\text{ALG}}\right]$; and

2. $G_j = \mathbb{E}\left[\widetilde{G}_j\right]$ for any $j \in A$.

Hence, to upper bound $\mathbb{E}\left[\text{REGRET}_i\right] = G_i - \mathbb{E}\left[G_{\text{ALG}}\right]$, it suffices to upper bound $\mathbb{E}\left[\widetilde{G}_i - \widetilde{G}_{\text{ALG}}\right]$. By the definition of the probability that the algorithm picks each arm, i.e., $\tilde{\mathbf{p}}(t)$, we have:

$$\mathbb{E}\left[\widetilde{G}_{\text{ALG}}\right] \geq (1-\gamma) \textstyle\sum_{t \in [T]} \mathbf{p}(t) \cdot \tilde{\mathbf{g}}(t) \ .$$

Hence, we have that for any initial distribution $\mathbf{q}$ over A:

$$\begin{aligned}\sum_{j \in A} q_j \cdot \mathbb{E}\left[\widetilde{G}_j\right] - \mathbb{E}\left[\widetilde{G}_{\text{ALG}}\right] &\leq \ \mathbb{E}\left[\textstyle\sum_{j \in A} q_j \cdot \widetilde{G}_j - \sum_{t \in [T]} \mathbf{p}(t) \cdot \tilde{\mathbf{g}}(t)\right] + \tfrac{\gamma}{1-\gamma} \mathbb{E}\left[\widetilde{G}_{\text{ALG}}\right] \\ &\leq \ \mathbb{E}\left[\textstyle\sum_{j \in A} q_j \cdot \widetilde{G}_j - \sum_{t \in [T]} \mathbf{p}(t) \cdot \tilde{\mathbf{g}}(t)\right] + 2\gamma \mathbb{E}\left[\widetilde{G}_{\text{ALG}}\right] \ . \qquad (19)\end{aligned}$$

Next, we upper bound the 1st term on the RHS. Note that $\mathbf{p}(t)$'s are the probabilities of choosing experts by MSMW when the experts have rewards $\tilde{\mathbf{g}}(t)$'s. By Proposition 10, we have that for any benchmark distribution $\mathbf{q}$ over S, the Bandit-MSMW algorithm satisfies that:

$$\sum_{j \in A} q_j \cdot \widetilde{G}_j - \sum_{t \in [T]} \mathbf{p}(t) \cdot \tilde{\mathbf{g}}(t) \leq \eta \sum_{t \in [T]} \sum_{j \in A} \frac{p_j(t)}{c_j} \cdot \big(\tilde{g}_j(t)\big)^2 + \frac{1}{\eta} \sum_{j \in A} c_j \left(q_j \ln\big(\frac{q_j}{p_j(1)}\big) - q_j + p_j(1) \right) \ . \qquad (20)$$

For any $t \in [T]$ and any $j \in A$, by the definition of $\tilde{g}_j(t)$, it equals $\frac{g_j(t)}{\tilde{p}_j(t)}$ with probability $\tilde{p}_j(t)$, and equals 0 otherwise. Thus, if we fix the random coin flips in the first $t-1$ rounds and, thus, fix $\tilde{\mathbf{p}}(t)$, and take expectation over the randomness in round t, we have that:

$$\mathbb{E}\left[\frac{p_j(t)}{c_j} \cdot \big(\tilde{g}_j(t)\big)^2\right] = \frac{p_j(t)}{c_j} \cdot \tilde{p}_j(t) \cdot \left(\frac{g_j(t)}{\tilde{p}_j(t)}\right)^2 = \frac{p_j(t)}{\tilde{p}_j(t)} \frac{(g_j(t))^2}{c_j} \ .$$

Further note that since $\tilde{p}_j(t) \geq (1-\gamma) p_j(t)$, and $g_j(t) \leq c_j$, the above is upper bounded by $\frac{1}{1-\gamma} g_j(t) \leq 2 g_j(t)$. Putting together with (20), we have that for any $0 < \eta \leq \frac{\gamma}{n}$:

$$\begin{aligned}\mathbb{E}\left[\sum_{j \in A} q_j \cdot \widetilde{G}_j - \sum_{t \in [T]} \mathbf{p}(t) \cdot \tilde{\mathbf{g}}(t)\right] &\leq \ \eta \sum_{t \in [T]} \sum_{j \in A} 2 g_j(t) + \frac{1}{\eta} \sum_{j \in A} c_j \left(q_j \ln\big(\frac{q_j}{p_j(1)}\big) - q_j + p_j(1) \right) \\ &= \ 2\eta \sum_{j \in A} G_j + \frac{1}{\eta} \sum_{j \in A} c_j \left(q_j \ln\big(\frac{q_j}{p_j(1)}\big) - q_j + p_j(1) \right)\end{aligned}$$

Combining with (19), we have:

$$\sum_{j \in A} q_j \cdot \mathbb{E}\left[\widetilde{G}_j\right] - \mathbb{E}\left[\widetilde{G}_{\text{ALG}}\right] \leq 2\eta \sum_{j \in A} G_j + \frac{1}{\eta} \sum_{j \in A} c_j \left(q_j \ln\big(\frac{q_j}{p_j(1)}\big) - q_j + p_j(1) \right) + 2\gamma \mathbb{E}\left[\widetilde{G}_{\text{ALG}}\right]$$

Let $\mathbf{q} = (1-\gamma)\mathbf{1}_i + \frac{\gamma}{k}\mathbf{1}$. Recall that $\mathbf{p}(1) = (1-\gamma)\mathbf{1}_{i_{\min}} + \frac{\gamma}{k}\mathbf{1}$ (recall $i_{\min}$ is the arm with minimum range $c_{i_{\min}}$). Similar to the discussion for the expert problem in Section 2.5, the 2nd term on the RHS is upper bounded by $O\left(\frac{1}{\eta} \log\big(\frac{k}{\gamma}\big) \cdot c_i\right)$. Hence, we have:

$$\sum_{j \in A} q_j \cdot \mathbb{E}\left[\widetilde{G}_j\right] - \mathbb{E}\left[\widetilde{G}_{\text{ALG}}\right] \leq 2\eta \sum_{j \in A} G_j + O\left(\frac{1}{\eta} \log\big(\frac{k}{\gamma}\big) \cdot c_i\right) + 2\gamma \mathbb{E}\left[\widetilde{G}_{\text{ALG}}\right] \ . \qquad (21)$$

Further, the LHS is lower bounded as:

$$(1-\gamma)\mathbb{E}\left[\widetilde{G}_i\right] + \frac{\gamma}{k}\sum_{j\in A}\mathbb{E}\left[\widetilde{G}_j\right] - \mathbb{E}\left[\widetilde{G}_{\text{ALG}}\right] \geq (1-\gamma)\mathbb{E}\left[\widetilde{G}_i\right] - \mathbb{E}\left[\widetilde{G}_{\text{ALG}}\right] .$$

The lemma then follows by putting it back to (21) and rearranging terms. ■

3.3 Regret bounds for non-negative rewards - proof of Theorem 12

Proof [**of Theorem 12**] Letting $\gamma = \epsilon$ and $\eta = \frac{\gamma}{k} = \frac{\epsilon}{k}$ in Lemma 14, we get that the expected regret w.r.t. an action $i \in A$ is bounded by:

$$O\left(\epsilon \cdot G_i + \tfrac{\epsilon}{k}\textstyle\sum_{j\in A} G_j + c_i \cdot \tfrac{k}{\epsilon}\ln\left(\tfrac{k}{\epsilon}\right)\right) .$$

When $i = i^*$ (best arm), regret is bounded by $O\left(\epsilon \cdot G_{i^*} + c_i^* \cdot \frac{k}{\epsilon}\ln\left(\frac{k}{\epsilon}\right)\right)$, as desired.

For the regret w.r.t. an arbitrary action, note that $\mathbb{E}\left[G_{\text{ALG}}\right] \geq \frac{\gamma}{k}\sum_{j\in A} G_j$. Thus, the regret bound w.r.t. an action $i \in A$ in Lemma 14 is further upper bounded by:

$$O\left(\tfrac{1}{\eta}\log\left(\tfrac{k}{\gamma}\right)\cdot c_i + \left(\tfrac{\eta k}{\gamma} + \gamma\right)\cdot \mathbb{E}\left[\widetilde{G}_{\text{ALG}}\right]\right)$$

The theorem then follows by letting $\gamma = \epsilon$ and $\eta = \frac{\gamma^2}{k} = \frac{\epsilon^2}{k}$. ■

4. More Applications of Multi-scale Learning for Auctions and Pricing

In this section, we consider applying the multi-scale online learning framework, developed in Section 2 and Section 3, to design several other online auctions and pricings be the single buyer auction (discussed in Section 2.6). Besides the single buyer auction, the problems that we consider are as follows.

- **Online posted pricing:** The same as the online single buyer auction of Section 2.6, but in the bandit setting. The algorithm only learns the indicator function $\mathbf{1}(v(t) \geq p_t)$ where p_t is the price it picks in round t.
- **Online multi buyer auction:** The action set is the set of all "Myerson-type" mechanisms for n buyers, for some $n \in \mathbb{N}$. (See Definition 15.) The adversary picks a valuation vector $\mathbf{v}(t) \in [1,h]^n$ and the reward of a mechanism M is its revenue when the valuation of the buyers is given by $\mathbf{v}(t)$; this is denoted by $\text{REV}_M(\mathbf{v}(t))$. The algorithm sees the full vector of valuations $\mathbf{v}(t)$.

4.1 Auctions and pricing as multi-scale online learning problems

We now show how to reduce the above problems to special cases of multi-scale online learning.

Online multi buyer auction In multi buyer auctions, we consider the set of all discretized Myerson-type auctions as the action space. We start by defining Myerson-type auctions:

Definition 15 (Myerson-type auctions) *A* Myerson-type auction *is defined by n non-decreasing virtual value mappings $\phi_1, \ldots, \phi_n : [1, h] \mapsto [-\infty, h]$. Given a value profile $v_1, \ldots, v_n$, the item is given to the bidder j with the largest non-negative virtual value $\phi_j(v_j)$. Then, bidder j pays the minimum value that would keep him as the the winner.*

Myerson (1981) shows that when the bidders' values are drawn from independent (but not necessarily identical) distributions, the revenue-optimal auction is a Myerson-type auction. Devanur et al. (2016, Lemma 5) observe that to obtain a $1-\epsilon$ approximation, it suffices to consider the set of discretized Myerson-type auctions that treat each bidder's value as if it is equal to the closest power of $1+\epsilon$ from below. As a result, it suffices to consider the set of discretized Myerson-type auctions, each of which is defined by the virtual values of $(1+\epsilon)^j$'s, i.e., by $O(n \log h/\epsilon)$ real numbers $\phi_\ell((1+\epsilon)^j)$, for $\ell \in [n]$, and $0 \le j \le \lfloor \log_{1+\epsilon} h \rfloor$. Furthermore, first Elkind (2007) and later on Devanur et al. (2016); Gonczarowski and Nisan (2017) note that a discretized Myerson-type auction is in fact completely characterized by the total ordering of $\phi_\ell((1+\epsilon)^j)$'s;[7] their actual values do not matter. Indeed, both the allocation rule and the payment rule are determined by the ordering of virtual values. As a result, our action space is a finite set with at most $O((n \log h/\epsilon)!)$ actions. The range of an action, i.e., a discretized Myerson-type auction, is the largest price ever charged by the auction, i.e., the largest value v of the form $(1+\epsilon)^j$ such that there exists $\ell \in [n]$, $\phi_\ell(v) > \phi_\ell((1+\epsilon)^{-1}v)$.

4.2 Multiplicative/additive approximations

Similar to Section 2.6, we show how to get a multiplicative cum additive approximations for these problems with G_{MAX} as the benchmark. Recall the definition of G_{MAX} in Section 2.1 and let p^* be the best fixed price on hindsight, which is the price that achieves G_{MAX}.

Theorem 16 *There are algorithms for the online posted pricing and the online multi buyer auction problems that take as input a parameter $\epsilon > 0$, and satsify $G_{\text{ALG}} \ge (1-\epsilon)G_{\text{MAX}} - O(E)$, where respectively (for the two problems mentioned above)*

$$E = \frac{p^* \log h \log(\log h/\epsilon)}{\epsilon^2}, \quad \textit{and} \quad \frac{hn \log h \log(n \log h/\epsilon)}{\epsilon^2} \ .$$

Even if h is not known up front, we can still get the similar approximation guarantee for the online multi buyer auction with:

$$E = \frac{hn \log h \log(n \log h/\epsilon)}{\epsilon^2} \ .$$

We conjecture that our bound for the online posted pricing problem is tight up to logarithmic factors, and leave resolving this as an open problem. The second bound is not comparable to the best sample complexity for the multi buyer auction problem by

7. Cai et al. (2012) also generalizes this observation to multi-dimensional types.

Roughgarden and Schrijvers (2016); it is better than theirs for large ϵ (when $1/\epsilon \leq o(nh)$), and is worse for smaller ϵ (when $1/\epsilon \geq \omega(nh)$). Also, compare the first bound to the corresponding upper bound for the pricing problem by Blum and Hartline (2005), which is

$$\min\left\{\frac{h\log h\log\log h}{\epsilon^2}, \frac{h\log\log h}{\epsilon^3}\right\}.$$

Essentially, the main improvement over this result is that the additive term scales with the best price rather than h.

4.3 Proof of Theorem 16

Proof *Online posted pricing.* Recall the formulation of the problem as an online learning problem with bandit feedback in Section 4.1. This part then follows by Theorem 12 with $k = O(\log h/\epsilon)$ actions.

Online multi buyer auction. Recall the formulation of the problem as an online learning problem with full information in Section 4.1. The proof then follows by Theorem 1, where we let $\boldsymbol{\pi}$ be the uniform distribution over the $k = O((n\log h/\epsilon)!)$ actions, i.e., Myerson-type auctions.

When h is not known up front, similar to the proof of Theorem 11, we consider a hypothetical algorithm with countably infinite action space A as follows. For any $p = (1+\epsilon)^j$, $j \geq 0$, let the $k_p = O((n\log p/\epsilon)!)$ Myerson-type auctions for values in $[1, p]$ be in A; we assume these auctions treat any values greater than p as if they were p. Further, we choose the prior distribution $\boldsymbol{\pi}$ such that the probability mass of each auction for range $[1, p]$ is equal to $\frac{\epsilon(\epsilon+2)}{(1+\epsilon)^2} \cdot \frac{1}{p^2} \cdot \frac{1}{k_p}$. The approximation guarantee then follows by Theorem 1. To implement this algorithm, we use the same trick as in the proof Theorem 11 by running a modified algorithm that only considers auctions for all ranges $[1, p]$ where p is no larger than the highest value seen so far among all the buyers (i.e. a multi-buyer auction version of Algorithm 2). The rest of the proof that shows the revenue loss of this algorithm compared to the hypothetical algorithm is negligible is similar to the proof of Theorem 11 (and hence omitted for brevity).

■

4.4 Competing with δ-guarded benchmarks

For the single buyer auction/pricing problem, we define a δ-guarded benchmark, for any $\delta \in [0, 1]$. This benchmark is restricted to only those prices that sell the item in at least a δ fraction of the rounds.

$$G_{\text{MAX}}(\delta) := \max\left\{\textstyle\sum_{t=1}^{T} g_p(t) : p \in A, \sum_{t=1}^{T} \mathbf{1}(v_t \geq p) \geq \delta T\right\}.$$

As observed in Footnote 2, one can replace δ with $1/h$ and get the corresponding guarantees for G_{MAX} rather than $G_{\text{MAX}}(\delta)$. However, the main point of these results is to show a graceful improvement of the bounds as δ is chosen to be larger.

Multiple buyers: For the multi buyer auction problem, we define the δ-guarded benchmark as follows. For any sequence of value vectors $\mathbf{v}(1), \mathbf{v}(2), \ldots, \mathbf{v}(T)$, let $\bar{V}$ denote the largest value such that there are at least δT distinct $t \in [1:T]$ with $\max_{i\in[n]} v_i(t) \geq \bar{V}$. Define the δ-guarded benchmark to be

$$G_{\text{MAX}}(\delta) = \max_M \sum_{t=1}^T Rev_M \left(\min(\bar{V}\vec{\mathbf{1}}, \mathbf{v}(t))\right),$$

where the "min" is taken coordinate-wise, and the "max" is over all Myerson-type mechanisms. In other words, here is how we can describe the δ-guarded benchmark: for each Myerson-type auction M, after identifying the value cap $\bar{V}$, we cut all the values that are above $\bar{V}$ by this quantity, and then run M. The benchmark is then the revenue of the best Myerson-type auction under these modified values.

We focus on purely multiplicative approximation factors when competing with $G_{\text{MAX}}(\delta)$. In particular, for any given $\epsilon > 0$, we are interested in a $1-\epsilon$ approximation. We state our results in terms of the *convergence rate*. We say that $T(\epsilon, \delta)$ is the convergence rate of an algorithm if for all time horizon $T \geq T(\epsilon, \delta)$, we are guaranteed that $G_{\text{ALG}} \geq (1-\epsilon) G_{\text{MAX}}(\delta)$. Our main results are as follows.

Theorem 17 *There are algorithms for the online single buyer auction, online posted pricing, and the online multi buyer auction problems with convergence rates respectively of*

$$O\left(\frac{\log(\log h/\epsilon)}{\epsilon^2 \delta}\right), \quad O\left(\frac{\log h}{\epsilon^4 \delta}\right), \quad \text{and } O\left(\frac{n\log(1/\epsilon\delta)\log(n\log(1/\epsilon\delta)/\epsilon)}{\epsilon^3\delta} + \frac{\log(\log h/\epsilon)}{\epsilon^2\delta}\right).$$

Even if h is not known upfront, we can still get the following similar convergence rates for online single buyer auction and online multi buyer auction respectively:

$$O\left(\frac{\log(p^*/\epsilon)}{\epsilon^2\delta}\right), \quad \text{and} \quad O\left(\frac{n\log(1/\epsilon\delta)\log(n\log(1/\epsilon\delta)/\epsilon)}{\epsilon^3\delta} + \frac{\log(h/\epsilon)}{\epsilon^2\delta}\right).$$

Once again, we compare to the sample compexity bounds: our first is within a $\log\log h$ factor of the best sample complexity upper bound in Huang et al. (2015b). The lower bound for the online single buyer auction is $\Omega(\delta^{-1}\epsilon^{-2})$, which is also the best lower bound known for the pricing and the multi-buyer problem.[8] For the online posted pricing problem, we conjecture that the right dependence on ϵ should be ϵ^{-3}. No sample complexity bounds for the multi-buyer problem were known before; in fact we introduce the definition of a δ-guarded benchmark for this problem.

4.5 Proof of Theorem 17

Proof *Online single buyer auction.* By Theorem 1, letting $\boldsymbol{\pi}$ be the uniform distribution over the $k = O(\log h/\epsilon)$ actions, i.e., discretized prices, we have that for any price p (recall that $c_p = p$):

$$G_{\text{ALG}} \geq (1-\epsilon)\cdot G_p - O\left(\tfrac{\log(\log h/\epsilon)}{\epsilon}\cdot p\right).$$

8. Cole and Roughgarden (2014) show that at least a linear dependence on n is necessary when the values are drawn from a regular distribution, but as is, their lower bound needs unbounded valuations. The lower bound probably holds for "large enough h" but it is not clear if it holds for all h.

For the δ-guarded optimal price p^* (i.e., subject to selling in at least δT rounds), we have $G_{p^*} \geq \delta T \cdot p^*$. Therefore, when $T \geq O\left(\log(\log h/\epsilon)/\epsilon^2\delta\right)$, the additive term of the above approximation guarantee is at most $\epsilon \cdot G_{p^*}$. So the theorem holds.

The treatment for the case when h is not known up front is essentially the same as in Theorem 16 and Theorem 11. As a hypothetical algorithm useful for analysis, we consider an algorithm (similar to Algorithm 1) with a countably infinite action space comprising all prices of the form $(1+\epsilon)^j$, for $j \geq 0$. Then, let the prior distribution $\boldsymbol{\pi}$ be such that for any price $p = (1+\epsilon)^j$, $\pi_p = \epsilon(\epsilon+2)(1+\epsilon)^{-2(j+1)} = \frac{\epsilon(eps+2))}{(1+\epsilon)^2} \cdot \frac{1}{p^2}$. The rest of the proof and how to implement is the same as in the proof of Theorem 11 (i.e. Algorithm 2).

Online posted pricing. Recall the above formulation of the problem as an online learning problem with bandit feedback. By Theorem 12 with $k = O(\log h/\epsilon)$ actions, we have that for any price p:

$$G_{\text{ALG}} \geq (1-\epsilon) \cdot G_p - O\left(\frac{\log h \log(\log h/\epsilon)}{\epsilon^3} \cdot p\right) \ .$$

Again, for the δ-guarded optimal price p^* (i.e., subject to selling in at least δT rounds), we have $G_{p^*} \geq \delta T \cdot p^*$. Therefore, when $T \geq O\left(\log h \log\left(\log h/\epsilon\right)/\epsilon^4\delta\right)$, the additive term of the above approximation guarantee is at most $\epsilon \cdot G_{p^*}$. So the theorem holds.

Online multi buyer auction. Suppose i^* is the δ-guarded best Myerson-type auction. Recall that $\bar{V}$ is the largest value such that there are at least δT distinct $v(t)$'s with $\max_{\ell \in [n]} v_\ell(t) \geq \bar{V}$. So we may assume without loss of generality that i^* does not distinguish values greater than $\bar{V}$. Hence:

$$c_{i^*} \leq \bar{V} \ . \tag{22}$$

Further, note that running a second-price auction with anonymous reserve $\bar{V}$ is a Myerson-type auction (e.g., mapping values less than $\bar{V}$ to virtual value $-\infty$ and values greater than or equal to $\bar{V}$ to virtual value $\bar{V}$), and it gets revenue at least $\delta T \cdot \bar{V}$. So we have that:

$$G_{p^*} \geq \delta T \cdot \bar{V} \ . \tag{23}$$

Finally, the above implies that to obtain a $1-\epsilon$ approximation, it suffices to consider prices that are at least $\epsilon\delta\bar{V}$. Hence, it suffices to consider Myerson-type auctions that, for a given $\bar{V}$, do not distinguish among values greater than $\bar{V}$, and do not distinguish among values smaller than $\epsilon\delta\bar{V}$. There are $O(\log h/\epsilon)$ different values of $\bar{V}$. Further, given $\bar{V}$, there are only $O(\log(1/\epsilon\delta)/\epsilon)$ distinct values to be considered and, thus, there are at most $O((n\log(1/\epsilon\delta)/\epsilon)!)$ distinct Myerson-type auctions of this kind. Hence, the total number of distinct Myerson-type actions that we need to consider is at most:

$$k = O\left(\frac{\log h}{\epsilon} \cdot \left(\frac{n\log(1/\epsilon\delta)}{\epsilon}\right)!\right) \ .$$

Letting $\boldsymbol{\pi}$ be the uniform distribution over the k actions in Theorem 1, we have that (recall Eqn. (22)):

$$G_{\text{ALG}} \geq (1-\epsilon) \cdot G_{i^*} - O\left(\frac{n\log(1/\epsilon\delta)\log(n\log(1/\epsilon\delta)/\epsilon)}{\epsilon^2} + \frac{\log(\log h/\epsilon)}{\epsilon}\right) \cdot \bar{V} \ .$$

When $T \geq O\left(\frac{n \log(1/\epsilon\delta) \log(n \log(1/\epsilon\delta)/\epsilon)}{\epsilon^3\delta} + \frac{\log(\log h/\epsilon)}{\epsilon^2\delta}\right)$, the additive term of the above approximation guarantee is at most $\epsilon \cdot G_{i^*}$ due to Eqn. (23). So the theorem holds.

Again, the treatment for the case when h is not known up front is similar to that in Theorem 16. When h is not known up front, we consider a hypothetical algorithm with a countably infinite action space A as follows. For any $\bar{V} = (1+\epsilon)^j$, $j \geq 0$, let the $k' = O((n \log(1/\epsilon\delta)/\epsilon)!)$ Myerson-type auctions that do not distinguish among values greater than $\bar{V}$, and do not distinguish among values smaller than $\epsilon\delta\bar{V}$ be in A. Further, we choose the prior distribution $\boldsymbol{\pi}$ such that the probability mass of each Myerson-type auction for a given $\bar{V}$ is equal to $\frac{\epsilon}{1+\epsilon} \cdot \frac{1}{\bar{V}} \cdot \frac{1}{k'}$. The approximation guarantee then follows by Theorem 1 and essentially the same argument as the known h case. Implementation is similar to the proof of Theorem 16 and Theorem 11 (i.e. a multi-buyer auction version of Algorithm 2). The rest of the proof that shows the revenue loss of this algorithm compared to the hypothetical algorithm is negligible is similar to the proof of Theorem 16 (and hence omitted for brevity). ■

Remark Devanur et al. (2016) show that when the values are drawn from independent regular distributions, the ϵ-guarded optimal price is a $1-\epsilon$ approximation of the unguarded optimal price. So our convergence rate for the online multi buyer auction problem in Theorem 1 implies a $\tilde{O}(n\epsilon^{-4})$ sample complexity modulo a mild $\log\log h$ dependency on the range, almost matching the best known sample complexity upper bound for regular distributions.

5. Multi-scale Online Learning with Symmetric Range

In this section, we consider multi-scale online learning when the rewards are in a symmetric range, i.e. for all $i \in A$ and $t \in [T]$, $g_i(t) \in [-c_i, c_i]$. The standard analysis for the experts and the bandit problems holds even if the range of $g_i(t)$ is $[-c_i, c_i]$, instead of $[0, c_i]$. In contrast, there are subtle differences on the best achievable multi-scale regret bounds between the non-negative and the symmetric range, which we explore in this section. We look at both the full information and bandit setting, and prove action-specific regret upper bounds. We then prove a tight lower-bound in Section 5.3 for the full information case, and an almost tight lower-bound in Section 5.5 for the bandit setting.

5.1 Multi-scale regret bounds for symmetric ranges

We first show the following upper bound for the full information setting when the range is symmetric. This bound follows the same style of action-specific regret bounds as in Theorem 1. More detailed discussion on how the choice of initial distribution $\boldsymbol{\pi}$ affects the bound is deferred to the appendix, Section A.1 (recall that the initial distribution $\boldsymbol{\pi}$ is the distribution over actions that is used in the first round of Algorithm 1).

Theorem 18 *There exists an algorithm for the multi-scale experts problem with symmetric range that takes as input any distribution $\boldsymbol{\pi}$ over A, the ranges c_i, $\forall\, i \in A$, and a parameter $0 < \epsilon \leq 1$, and satisfies:*

$$\forall i \in A: \quad \mathbb{E}\left[\text{REGRET}_i\right] \leq \epsilon \cdot \mathbb{E}\left[\sum_{t \in [T]} \left|g_t(i)\right|\right] + O\left(\frac{1}{\epsilon} \log\left(\frac{1}{\pi_i} \cdot \frac{c_i}{c_{\min}}\right) \cdot c_i\right). \tag{24}$$

Similar to Section 2.1, we can compute the pure-additive version of the bound in Theorem 18 by setting $\epsilon = \sqrt{\frac{\log(k\cdot\frac{c_{\max}}{c_{\min}})}{T}}$, as in Corollary 2.

Corollary 19 *There exists an algorithm for the online multi-scale experts problem with symmetric range that takes as input the ranges c_i, $\forall\ i \in A$, and satisfies:*

$$\forall i \in A: \quad \mathbb{E}\left[\text{REGRET}_i\right] \le O\left(c_i \cdot \sqrt{T\log(k\cdot\tfrac{c_{\max}}{c_{\min}})}\right) \tag{25}$$

If we compare the above regret bound with the standard $O(c_{\max}\sqrt{T\log k})$ regret bound for the experts problem, we see that we replace the dependency on $c_{\max}$ in the standard bound with $c_i\sqrt{\log(\frac{c_{\max}}{c_{\min}})}$. It is natural to ask whether we could get rid of the dependence on $\log(c_i/c_{\min})$ and show a regret bound of $O(c_i\sqrt{T\log k})$, like we did for non-negative rewards. However, the next theorem shows that this dependence on $\log(c_i/c_{\min})$ in the above bound is necessary, in a weak sense: where the constant in the $O(\cdot)$ is universal and does not depend on the ranges c_i. This is because the lower bound only holds for "small" values of the horizon T, which nonetheless grows with the $\{c_i\}$s.[9]

Theorem 20 *There exists an action set of size k, and ranges $c_i, \forall i \in [k]$, and time horizon T, such that for all algorithms for the online multi-scale experts problem with symmetric range, there is a sequence of T gain vectors such that*

$$\exists i \in A: \quad \mathbb{E}\left[\text{REGRET}_i\right] > \tfrac{c_i}{4} \cdot \sqrt{T\log(k\cdot\tfrac{c_{\max}}{c_{\min}})}$$

We then show the following upper bound for the bandit setting when the range is symmetric. This bound also follows the same style of action-specific regret bounds as in Theorem 12.

Theorem 21 *There exists an algorithm for the multi-scale bandits problem with symmetric range that takes as input the ranges c_i, $\forall\ i \in A$, and a parameter $0 < \epsilon \le 1/2$, and satisfies:*

$$\forall i \in A: \quad \mathbb{E}\left[\text{REGRET}_i\right] \le O\big(\epsilon T + \tfrac{k}{\epsilon}\tfrac{c_{\max}}{c_{\min}}\log\big(\tfrac{k}{\epsilon}\tfrac{c_{\max}}{c_{\min}}\big)\big)\cdot c_i. \tag{26}$$

Also, similar to Section 2.1, we can compute the pure-additive version of the bound in Theorem 21 by setting $\epsilon = \sqrt{\frac{k\frac{c_{\max}}{c_{\min}}\log(kT\cdot\frac{c_{\max}}{c_{\min}})}{T}}$, as in Corollary 2. This bound is comparable to the standard regret bound of $O(c_{\max}\sqrt{kT\log k})$ (Auer et al., 1995) for the adversarial multi-armed bandits problem.

Corollary 22 *There exists an algorithm for the online multi-scale bandits problem with symmetric range that satisfies:*

$$\forall i \in A: \quad \mathbb{E}\left[\text{REGRET}_i\right] \le O\left(c_i \cdot \sqrt{Tk\cdot\tfrac{c_{\max}}{c_{\min}}\log(kT\cdot\tfrac{c_{\max}}{c_{\min}})}\right). \tag{27}$$

Once again, for the bandit problem, the following theorem shows that this bound cannot be improved beyond logarithmic factors (to get a guarantee like that of Theorem 12, for instance).

9. For this reason we chose not to include this bound in Table 1.

Theorem 23 *There exists an action set of size k, and ranges $c_i, \forall i \in [k]$, such that for all algorithms for the online multi-scale bandit problem with symmetric range, for all sufficiently large time horizon T, there is a sequence of T gain vectors such that*

$$\exists i \in A: \quad \mathbb{E}\left[\text{REGRET}_i\right] > \frac{c_i}{8\sqrt{2}} \cdot \sqrt{Tk \cdot \frac{c_{\max}}{c_{\min}}}.$$

5.2 Upper bound for experts with symmetric range - Proof of Theorem 18

Recall the proof of Proposition 10. The proof only requires $g_i(t) \in [-c_i, c_i]$ for all $i \in A, t \in [T]$. Choosing q to be $\mathbf{1}_i$, a vector with a 1-entry in i^{th} coordinate and 0-entries elsewhere for an action $i \in A$, and noting that

$$\textstyle\sum_{t\in[T]}\sum_{i\in A} p_i(t)\frac{(g_i(t))^2}{c_i} \leq \sum_{t\in[T]}\sum_{i\in A} p_i(t) \cdot \left|g_i(t)\right| \ ,$$

we get the following regret bound as a corollary of Proposition 10.

Corollary 24 *For any initial distribution μ over A, and any learning rate parameter $0 < \eta \leq 1$, the MSMW algorithm achieves the following regret bound:*

$$\forall i \in A: \quad \mathbb{E}\left[\text{REGRET}_i\right] \leq \eta \cdot \mathbb{E}\left[\textstyle\sum_{t\in[T]} \left|g_i(t)\right|\right] + \frac{1}{\eta} c_i \cdot \log\left(\frac{1}{\mu_i}\right) + \frac{1}{\eta}\textstyle\sum_{j\in A}\mu_j c_j \tag{28}$$

Now, we can prove the multi-scale regret upper-bound in Theorem 18 using Corollary 24.
Proof [of Theorem 18] The proof follows by choosing an appropriate initial distribution μ in Corollary 24. By Corollary 24, we have:

$$\mathbb{E}\left[\text{REGRET}_i\right] \leq \eta \cdot \mathbb{E}\left[\textstyle\sum_{t\in[T]} \left|g_i(t)\right|\right] + \frac{1}{\eta} c_i \cdot \log(\frac{1}{\mu_i}) + \frac{1}{\eta}\textstyle\sum_{j\in A}\mu_j c_j$$

Let $i_{\min}$ be an action with the minimum range $c_{i_{\min}} = c_{\min}$. Consider an initial distribution $\mu_j = \pi_j \frac{c_{\min}}{c_j}$ for all $j \neq i_{\min}$, and $\mu_{i_{\min}} = 1 - \sum_{j\neq i_{\min}} \mu_j$, i.e., putting all remaining probability mass on action $i_{\min}$. Then, the third term on the RHS is upper bounded by:

$$\textstyle\sum_{j\in A}\mu_j c_j = \sum_{j\neq i_{\min}}\mu_j c_j + \mu_{i_{\min}} c_{i_{\min}} = \sum_{j\neq i_{\min}}\pi_j c_{\min} + \mu_{i_{\min}} c_{\min} \leq 2c_{\min} \leq 2c_i \ \ .$$

For $i \neq i_{\min}$, by the definition of μ_i, we have:

$$\begin{aligned}\mathbb{E}\left[\text{REGRET}_i\right] &\leq \eta \cdot \mathbb{E}\left[\textstyle\sum_{t\in[T]} \left|g_i(t)\right|\right] + \frac{1}{\eta} c_i \cdot \log(\frac{1}{\pi_i} \cdot \frac{c_i}{c_{\min}}) + \frac{1}{\eta} \cdot 2c_{\min} \\ &= \eta \cdot \mathbb{E}\left[\textstyle\sum_{t\in[T]} \left|g_i(t)\right|\right] + O\left(\frac{1}{\eta}\log\left(\frac{1}{\pi_i} \cdot \frac{c_i}{c_{\min}}\right) \cdot c_i\right) \ .\end{aligned}$$

So the theorem follows by choosing $\eta = \epsilon$. For $i = i_{\min}$, note that $\mu_j \leq \pi_j$ for all $j \neq i_{\min}$ and, thus, $\mu_{i_{\min}} = 1 - \sum_{j\neq i_{\min}} \mu_j \geq 1 - \sum_{j\neq i_{\min}} \pi_j = \pi_{i_{\min}} = \pi_{i_{\min}} \frac{c_{\min}}{c_{i_{\min}}}$. The theorem then holds following the same calculation as in the $j \neq i_{\min}$ case. ■

5.3 Lower bound for experts with symmetric range - proof of Theorem 20

Proof [**of Theorem 20**] We first show that for any online learning algorithm, and any sufficiently large $h > 1$, there is an instance that has two experts with $c_1 = 1$ and $c_2 = h$ with $T = \Theta(\log h)$ rounds, such that either

$$\mathbb{E}\left[\text{REGRET}_1\right] > \tfrac{1}{2}T + \sqrt{h}\ , \qquad \text{or} \qquad \mathbb{E}\left[\text{REGRET}_2\right] > \tfrac{1}{2}Th + \tfrac{1}{5}h\log_2 h\ .$$

We will construct this instance with $T = \frac{1}{2}\log_2 h - 1$ rounds adaptively that always has gain 0 for action 1 and gain either h or $-h$ for action 2. The proof of the theorem then follows as $c_{\min} = 1$, $c_{\max} = h$, $T = \frac{1}{2}\log_2 h - 1$, and $k = 2$ in this instance. Let q_t denote the probability that the algorithm picks action 2 in round t after having the same rewards 1 and h for the two actions respectively in the first $t-1$ rounds. We will first show that (1) if the algorithm has small regret with respect to action 1, then q_t must be upper bounded since the adversary may let action 2 have cost $-h$ in any round t in which q_t is too large. Then, we will show that (2) since q_t is upper bounded for any $1 \le t \le T$, the algorithm must have large regret with respect to action 2.

We proceed with the upper bounding q_t's. Concretely, we will show the following lemma.

Lemma 25 *Suppose* $\mathbb{E}\left[\text{REGRET}_1\right] \le \frac{1}{2}T + \sqrt{h}$. *Then, for any* $1 \le t \le T$, *we have* $q_t \le \frac{2^t}{\sqrt{h}}$.

Proof [Proof of Lemma 25] We will prove by induction on t. Consider the base case $t = 1$. Suppose for contradiction that $q_1 > \frac{2}{\sqrt{h}}$. Then, consider an instance in which action 2 always has gain. In this case, the expected gain of the algorithm (even if it always correctly picks action 1 in the remaining instance) is at most $q_1 \cdot (-h) < -2\sqrt{h}$. This is a contradiction to the assumption that $\mathbb{E}\left[\text{REGRET}_1\right] \le \frac{1}{2}T + \sqrt{h} < 2\sqrt{h}$.

Next, suppose the lemma holds for all rounds prior to round t. Then, the expected gain of algorithm in the first $t-1$ rounds if arm 2 has gain H is

$$\sum_{\ell=1}^{t-1} q_\ell \cdot h \le \sum_{\ell=1}^{t-1} 2^\ell \sqrt{h} = \left(2^t - 2\right)\sqrt{h}\ .$$

Suppose for contradiction that $q_t > \frac{2^t}{\sqrt{h}}$. Then, consider an instance in which action 2 has gain H in the first $t-1$ rounds and $-H$ afterwards. In this case, the expected gain of the algorithm (even if it always correctly picks action 1 after round t) is at most

$$\left(2^t - 2\right)\sqrt{h} + q_t(-h) < \left(2^t - 2\right)\sqrt{h} + 2^t\sqrt{h} < -2\sqrt{h}\ .$$

This is a contradiction to the assumption that $\mathbb{E}\left[\text{REGRET}_1\right] \le \frac{1}{2}T + \sqrt{h} < 2\sqrt{h}$. ■

Consider an instance in which action 2 always has gain H. Suppose that $\mathbb{E}\left[\text{REGRET}_1\right] \le \frac{1}{2}T + \sqrt{h}$. As an immediate implication of the above lemma, the algorithm is that the expected gain of the algorithm is upper bounded by:

$$\sum_{t=1}^{T} q_t h \le \sum_{t=1}^{T} 2^t \sqrt{h} < 2^{T+1}\sqrt{h} = h\ .$$

Note that in this instance $\mathbb{E}[G_2] = T \cdot h$. Thus, the regret w.r.t. action 2 is at least $(T-1)h$, which is greater than $\frac{1}{2} \cdot \mathbb{E}[G_2] + \frac{1}{5} h \log_2 h$ for sufficiently large h. ■

5.4 Upper bound for bandits with symmetric range - Proof of Theorem 21

We start by presenting the following regret bound, whose proof is an alteration of that for Lemma 14 under symmetric range. Next, we prove Theorem 21.

Lemma 26 *For any exploration rate $0 < \gamma \le \min\{\frac{1}{2}, \frac{c_{\min}}{c_{\max}}\}$ and any learning rate $0 < \eta \le \frac{\gamma}{k}$, the Bandit-MSMW algorithm (Algorithm 3) achieves the following regret bound:*

$$\forall i \in A : \mathbb{E}[\text{REGRET}_i] \le O\left(\frac{1}{\eta} \log\left(\frac{k}{\gamma}\right) \cdot c_i + \gamma T \cdot c_{\max}\right)$$

Proof [of Lemma 26] We further define:

$$\begin{aligned} \widetilde{G}_{\text{ALG}} &\triangleq \textstyle\sum_{t\in[T]} g_{i_t}(t) = \sum_{t\in[T]} \tilde{\mathbf{p}}(t) \cdot \tilde{\mathbf{g}}(t) \ , \\ \widetilde{G}_j &\triangleq \textstyle\sum_{t\in[T]} \tilde{g}_j(t) \ . \end{aligned}$$

In expectation over the randomness of the algorithm, we have:

1. $\mathbb{E}[G_{\text{ALG}}] = \mathbb{E}\left[\widetilde{G}_{\text{ALG}}\right]$; and
2. $G_j = \mathbb{E}\left[\widetilde{G}_j\right]$ for any $j \in A$.

Hence, to upper bound $\mathbb{E}[\text{REGRET}_i] = G_i - \mathbb{E}[G_{\text{ALG}}]$, it suffices to upper bound $\mathbb{E}\left[\widetilde{G}_i - \widetilde{G}_{\text{ALG}}\right]$.

By the definition of the probability that the algorithm picks each arm, i.e., $\tilde{\mathbf{p}}(t)$, and that reward of each round is at least $-c_{\max}$, we have that:

$$\mathbb{E}\left[\widetilde{G}_{\text{ALG}}\right] \ge (1-\gamma) \sum_{t\in[T]} \mathbf{p}(t) \cdot \tilde{\mathbf{g}}(t) - \gamma T c_{\max} \ .$$

Hence, for any benchmark distribution $\mathbf{q}$ over A, we have that:

$$\begin{aligned} \textstyle\sum_{j\in A} q_j \cdot \mathbb{E}\left[\widetilde{G}_j\right] - \mathbb{E}\left[\widetilde{G}_{\text{ALG}}\right] &\le \mathbb{E}\left[\textstyle\sum_{j\in A} q_j \cdot \widetilde{G}_j - \sum_{t\in[T]} \mathbf{p}(t) \cdot \tilde{\mathbf{g}}(t)\right] + \tfrac{\gamma}{1-\gamma}\mathbb{E}\left[\widetilde{G}_{\text{ALG}}\right] + \tfrac{\gamma}{1-\gamma} T c_{\max} \\ &\le \mathbb{E}\left[\textstyle\sum_{j\in A} q_j \cdot \widetilde{G}_j - \sum_{t\in[T]} \mathbf{p}(t) \cdot \tilde{\mathbf{g}}(t)\right] + 2\gamma\mathbb{E}\left[\widetilde{G}_{\text{ALG}}\right] + 2\gamma T c_{\max} \\ &\le \mathbb{E}\left[\textstyle\sum_{j\in A} q_j \cdot \widetilde{G}_j - \sum_{t\in[T]} \mathbf{p}(t) \cdot \tilde{\mathbf{g}}(t)\right] + 4\gamma T c_{\max} \ . \end{aligned} \tag{29}$$

where the 2nd inequality is due to $\gamma \le \frac{1}{2}$, and the 3rd inequality follows by that $c_{\max}$ is the largest possible reward per round.

Next, we upper bound the 1st term on the RHS of (29). Note that $\mathbf{p}(t)$'s are the probability of choosing experts by MSMW when the experts have rewards $\tilde{\mathbf{g}}(t)$'s. By Proposition 10, we have that for any benchmark distribution $\mathbf{q}$ over S, the Bandit-MSMW algorithm satisfies that:

$$\sum_{j\in A} q_j\cdot\widetilde{G}_j - \sum_{t\in[T]} \mathbf{p}(t)\cdot\tilde{\mathbf{g}}(t) \le \eta \sum_{t\in[T]}\sum_{j\in A} \frac{p_j(t)}{c_j}\cdot\left(\tilde{g}_j(t)\right)^2 + \frac{1}{\eta}\sum_{j\in A} c_j \left(q_j \ln\left(\frac{q_j}{p_j(1)}\right) - q_j + p_j(1)\right) . \tag{30}$$

For any $t\in[T]$ and any $j\in A$, by the definition of $\tilde{g}_j(t)$, it equals $\frac{g_j(t)}{\tilde{p}_j(t)}$ with probability $\tilde{p}_j(t)$, and equals 0 otherwise. Thus, if we fix the random coin flips in the first $t-1$ rounds and, thus, fix $\tilde{\mathbf{p}}(t)$, and take expectation over the randomness in round t, we have that:

$$\mathbb{E}\left[\frac{p_j(t)}{c_j}\cdot\left(\tilde{g}_j(t)\right)^2\right] = \frac{p_j(t)}{c_j}\cdot\tilde{p}_j(t)\cdot\left(\frac{g_j(t)}{\tilde{p}_j(t)}\right)^2 = \frac{p_j(t)}{\tilde{p}_j(t)}\frac{(g_j(t))^2}{c_j} .$$

Further note that $\tilde{p}_j(t)\ge(1-\gamma)p_j(t)$, and $|g_j(t)|\le c_j$, the above is upper bounded by $\frac{1}{1-\gamma}|g_j(t)|\le 2|g_j(t)|\le 2c_{\max}$. Putting together with (30), we have that for any $0<\eta\le\frac{\gamma}{n}$:

$$\begin{aligned}\mathbb{E}\left[\sum_{j\in A} q_j\cdot\widetilde{G}_j - \sum_{t\in[T]}\mathbf{p}(t)\cdot\tilde{\mathbf{g}}(t)\right] &\le \eta\sum_{t\in[T]}\sum_{j\in A} 2c_{\max} + \frac{1}{\eta}\sum_{j\in A} c_j\left(q_j\ln\left(\frac{q_j}{p_j(1)}\right) - q_j + p_j(1)\right)\\ &= 2\eta T k c_{\max} + \frac{1}{\eta}\sum_{j\in A} c_j\left(q_j\ln\left(\frac{q_j}{p_j(1)}\right) - q_j + p_j(1)\right)\end{aligned}$$

Combining with (29), we have (recall that $\eta\le\frac{\gamma}{k}$):

$$\begin{aligned}\sum_{j\in A} q_j\cdot\mathbb{E}\left[\widetilde{G}_j\right] - \mathbb{E}\left[\widetilde{G}_{\text{ALG}}\right] &\le 2\eta T k c_{\max} + \frac{1}{\eta}\sum_{j\in A} c_j\left(q_j\ln\left(\frac{q_j}{p_j(1)}\right) - q_j + p_j(1)\right) + 4\gamma T c_{\max}\\ &\le \frac{1}{\eta}\sum_{j\in A} c_j\left(q_j\ln\left(\frac{q_j}{p_j(1)}\right) - q_j + p_j(1)\right) + 6\gamma T c_{\max}\end{aligned}$$

Let $\mathbf{q}=(1-\gamma)\mathbf{1}_i+\frac{\gamma}{k}\mathbf{1}$. Recall that $\mathbf{p}(1)=(1-\gamma)\mathbf{1}_{i_{\min}}+\frac{\gamma}{k}\mathbf{1}$ (recall $i_{\min}$ is the arm with minimum range $c_{i_{\min}}$). Similar to the discussion for the expert problem in Section 2.5, the 1st term on the RHS is upper bounded by $O\left(\frac{1}{\eta}\log\left(\frac{k}{\gamma}\right)\cdot c_i\right)$. Hence, we have:

$$\sum_{j\in A} q_j\cdot\mathbb{E}\left[\widetilde{G}_j\right] - \mathbb{E}\left[\widetilde{G}_{\text{ALG}}\right] \le O\left(\frac{1}{\eta}\log\left(\frac{k}{\gamma}\right)\cdot c_i\right) + 6\gamma T c_{\max} . \tag{31}$$

Further, the LHS is lower bounded as:

$$(1-\gamma)\mathbb{E}\left[\widetilde{G}_i\right] + \frac{\gamma}{k}\sum_{j\in A}\mathbb{E}\left[\widetilde{G}_j\right] - \mathbb{E}\left[\widetilde{G}_{\text{ALG}}\right] \ge (1-\gamma)\mathbb{E}\left[\widetilde{G}_i\right] - \gamma T c_{\max} - \mathbb{E}\left[\widetilde{G}_{\text{ALG}}\right] .$$

The lemma then follows by putting it back to (31) and rearranging terms. ∎

Proof [**of Theorem 21**] Let $\gamma=\epsilon\frac{c_{\min}}{c_{\max}}$ and $\eta=\frac{\gamma}{k}$ in Lemma 26. Theorem follows noting that $\gamma c_{\max}=\epsilon c_{\min}\le\epsilon c_i$. ∎

5.5 Lower-bound for bandits with symmetric range - Proof of Theorem 23

Proof [of Theorem 23] We first show that for any online multi-scale bandits algorithm problem, and there is an instance that has two arms with $c_1 = 1$ and $c_2 = h$ for some sufficiently large h, a sufficiently large T, and $\epsilon = \sqrt{\frac{h}{256T}}$, such that either

$$\mathbb{E}\left[\text{REGRET}_1\right] > \epsilon T + \tfrac{1}{256\epsilon} h \ , \qquad \text{or} \qquad \mathbb{E}\left[\text{REGRET}_2\right] > \epsilon T h + \tfrac{1}{256\epsilon} h^2$$

We will prove the existence of this instance by looking at the stochastic setting, i.e., the gain vectors $\mathbf{g}(t)$'s are i.i.d. for $1 \le t \le T$. We consider two instances, both of which admit a fixed gain of 0 for action 1. In the first instance, the gain of action 2 is h with probability $\frac{1}{2} - 2\epsilon$, and $-h$ otherwise. Hence, the expected gain of playing action 2 is $-4\epsilon h$ per round in instance 1. In the second instance, the gain of action 2 is h with probability $\frac{1}{2} + 2\epsilon$, and $-h$ otherwise. Hence, the expected gain of playing action two is $4\epsilon h$ per round in instance 2. Note this proves the theorem, as $c_{\min} = 1$, $c_{\max} = h$, $k = 2$ and and $T = \frac{h}{256\epsilon^2}$.

Suppose for contradiction that the algorithm satisfies:

$$\mathbb{E}\left[\text{REGRET}_1\right] \le \epsilon T + \tfrac{1}{256\epsilon} h = \tfrac{1}{128\epsilon} h \quad , \quad \mathbb{E}\left[\text{REGRET}_2\right] \le \epsilon h T + \tfrac{1}{256\epsilon} h^2 = \tfrac{1}{128\epsilon} h^2 \ .$$

Let N_1 denote the expected number of times that the algorithm plays action 2 in instance 1. Then, the expected regret with respect to action 1 in instance 1 is $N_1 \cdot 4\epsilon h$. By the assumption that $\mathbb{E}\left[\text{REGRET}_1\right] \le \frac{1}{128\epsilon} h$, we have $N_1 \le \frac{1}{512\epsilon^2}$.

Next, by standard calculation, we get that the Kullback-Leibler (KL) divergence of the observed rewards in a single round in the two instances is 0 if action 1 is played and is at most $64\epsilon^2$ (for $0 < \epsilon < 0.1$) if action 2 is played. So the KL divergence of the observed reward sequences in the two instances is at most $64\epsilon^2 \cdot N_1 \le \frac{1}{8}$.

Then, we use a standard inequality about KL divergences. For any measurable function $\psi : X \mapsto \{1, 2\}$, we have $\Pr_{X \sim \rho_1}\big(\psi(X) = 2\big) + \Pr_{X \sim \rho_2}\big(\psi(X) = 1\big) \ge \frac{1}{2} \exp\big(-KL(\rho_1, \rho_2)\big)$. For any $1 \le t \le T$, let ρ_1 and ρ_2 be the distribution of observed rewards up to a round t in the two instances, and let $\psi(X)$ be the action played by the algorithm. By this inequality and the above bound on the KL divergence between the observed rewards in the two instances, we get that in each round, the probability that the algorithm plays action 2 in instance 1, plus the probability that the algorithm plays action 1 in instance 2, is at least $\frac{1}{2} \exp\left(-\frac{1}{8}\right) > \frac{2}{5}$ in any round t. Thus, the expected number of times that the algorithm plays action 1 in instance 2 from round 1 to T, denoted as N_2, is at least $N_2 \ge \frac{2}{5} \cdot T - N_1 \ge \frac{1}{3} \cdot T$, where the second inequality holds for sufficiently large h. Therefore, the expected regret w.r.t. action 2 in instance 2 is at least: $4\epsilon h \cdot \frac{1}{3} \cdot T = \frac{4}{3} \epsilon h T > \frac{1}{128\epsilon} h^2$. This is a contradiction to our assumption that $\mathbb{E}\left[\text{REGRET}_2\right] \le \frac{1}{128\epsilon} h^2$. ■

6. Conclusion

Revenue management has emerged as a competitive toolbox of strategies for increasing the profit of web-based markets. In particular, dynamic pricing, and dynamic auction design as its less mature relative, have become prevalent market mechanisms in nearly all industries. In this paper, we studied these problems from the perspective of online learning. For the online auction for single buyer, we showed regret bounds that scale with the best fixed price, rather than the range of the values (with a generalization to learning auctions). Moreover, we demonstrated a connection between the optimal regret bounds for this problem and offline sample complexity lower-bounds of approximating optimal revenue, studied in Cole and Roughgarden (2014); Huang et al. (2015a). Using this connection, we showed our regret bounds are almost optimal as they match these information theoretic lower-bounds. We further generalized our result to online pricing (bandit feedback) and online auction with multiple-buyers.

The key to our development and improved regret bounds for online auction design is generalizing the classical learning from experts and multi-armed bandit problems to their "multi-scale versions", where the reward of each action is in a different range. Here the objective is to design online learning algorithms whose regret with respect to a given action scales with its own range, rather than the maximum range. We showed how a variant of online mirror descent solves this learning problem.

Acknowledgments

The authors would like to thank Bobby Kleinberg, Dylan Foster, and Aranyak Mehta for helpful discussions about this work. We would also like to thank anonymous reviewers for their constructive and helpful comments.

References

Shipra Agrawal and Nikhil R Devanur. Bandits with concave rewards and convex knapsacks. In *Proceedings of the fifteenth ACM conference on Economics and computation*, pages 989–1006. ACM, 2014.

Kareem Amin, Afshin Rostamizadeh, and Umar Syed. Learning prices for repeated auctions with strategic buyers. In *Advances in Neural Information Processing Systems*, pages 1169–1177, 2013.

Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012.

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

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

Moshe Babaioff, Shaddin Dughmi, Robert Kleinberg, and Aleksandrs Slivkins. Dynamic pricing with limited supply. *ACM Transactions on Economics and Computation*, 3(1):4, 2015.

Ashwinkumar Badanidiyuru, Robert Kleinberg, and Aleksandrs Slivkins. Bandits with knapsacks. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 207–216. IEEE, 2013.

Maria-Florina Balcan, Avrim Blum, Jason D Hartline, and Yishay Mansour. Reducing mechanism design to algorithm design via machine learning. *Journal of Computer and System Sciences*, 74(8):1245–1270, 2008.

Ziv Bar-Yossef, Kirsten Hildrum, and Felix Wu. Incentive-compatible online auctions for digital goods. In *Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 964–970. Society for Industrial and Applied Mathematics, 2002.

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.

Avrim Blum and Jason D Hartline. Near-optimal online auctions. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1156–1163. Society for Industrial and Applied Mathematics, 2005.

Avrim Blum, Vijay Kumar, Atri Rudra, and Felix Wu. Online learning in online auctions. *Theoretical Computer Science*, 324(2-3):137–146, 2004.

Sébastien Bubeck. Introduction to online optimization. *Lecture Notes*, pages 1–86, 2011.

Yang Cai, Constantinos Daskalakis, and S Matthew Weinberg. Optimal multi-dimensional mechanism design: Reducing revenue to welfare maximization. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 130–139. IEEE, 2012.

Richard Cole and Tim Roughgarden. The sample complexity of revenue maximization. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 243–252, 2014.

Arnoud V den Boer. Dynamic pricing and learning: historical origins, current research, and new directions. *Surveys in operations research and management science*, 20(1):1–18, 2015.

Nikhil R Devanur, Zhiyi Huang, and Christos-Alexandros Psomas. The sample complexity of auctions with side information. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*, pages 426–439. ACM, 2016.

Peerapong Dhangwatnotai, Tim Roughgarden, and Qiqi Yan. Revenue maximization with a single sample. *Games and Economic Behavior*, 2014.

Edith Elkind. Designing and learning optimal finite support auctions. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 736–745. Society for Industrial and Applied Mathematics, 2007.

Dylan J Foster, Satyen Kale, Mehryar Mohri, and Karthik Sridharan. Parameter-free online learning via model selection. In *Advances in Neural Information Processing Systems*, pages 6020–6030, 2017.

Yoav Freund and Robert E Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pages 23–37. Springer, 1995.

Yannai A Gonczarowski and Noam Nisan. Efficient empirical revenue maximization in single-parameter auction environments. In *Proceedings of the ACM STOC*, 2017.

Zhiyi Huang, Yishay Mansour, and Tim Roughgarden. Making the most of your samples. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation, EC '15, Portland, OR, USA, June 15-19, 2015*, pages 45–60, 2015a.

Zhiyi Huang, Yishay Mansour, and Tim Roughgarden. Making the most of your samples. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, pages 45–60. ACM, 2015b.

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.

Jamie H Morgenstern and Tim Roughgarden. On the pseudo-dimension of nearly optimal auctions. In *Advances in Neural Information Processing Systems*, pages 136–144, 2015.

Roger B. Myerson. Optimal auction design. *Mathematics of Operations Research*, 6(1): 58–73, 1981.

Tim Roughgarden and Okke Schrijvers. Ironing in the dark. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, pages 1–18. ACM, 2016.

Ilya Segal. Optimal pricing mechanisms with unknown demand. *The American economic review*, 93(3):509–529, 2003.

Vasilis Syrgkanis. A sample complexity measure with applications to learning optimal auctions. *arXiv preprint arXiv:1704.02598*, 2017.

Kalyan T Talluri and Garrett J Van Ryzin. *The theory and practice of revenue management*, volume 68. Springer Science & Business Media, 2006.

Appendix A. Other Deferred Proofs and Discussions

A.1 Discussion on choice of $\boldsymbol{\pi}$ for bandit symmetric range

We now describe how the choice of initial distribution $\boldsymbol{\pi}$ affects the bound given in Theorem 18.

- When the action set is finite, we can choose $\boldsymbol{\pi}$ to be the uniform distribution to get the term
$$O\big(\frac{1}{\epsilon}\log(kc_i/c_{\min})\cdot c_i\big)$$
This recovers the standard bound by setting $c_i = c_{\max}$ for all $i \in A$.

- We can choose $\pi_i = \frac{c_i}{\sum_{j\in A} c_j}$ to get $O\big(\frac{1}{\epsilon}\log(\sum_{j\in A} c_j/c_{\min})\cdot c_i\big)$. In particular, if the c_i's form an arithmetic progression with a constant difference then this is just $O\big(\frac{\log k}{\epsilon}\cdot c_i\big)$.

A.2 Proof of Proposition 10 from first principles

We also provide an elementary proof of this lemma using first principles.

Proof [of Proposition 10] Based on the update rule of Algorithm 1, we have $g_i(t) = \frac{c_i}{\eta}\log(\frac{w_i(t+1)}{p_i(t)})$ for any $i \in A$. Therefore:

$$\begin{aligned}
\mathbf{g}(t)\cdot\big(\mathbf{q}-\mathbf{p}(t)\big) &= \sum_{i\in A} g_i(t)\big(q_i - p_i(t)\big) \\
&= \sum_{i\in A}\frac{c_i}{\eta}\cdot\log\big(\frac{w_i(t+1)}{p_i(t)}\big)\cdot\big(q_i-p_i(t)\big) \\
&= \frac{1}{\eta}\left(\sum_{i\in S} c_i\cdot q_i\cdot\log\big(\frac{w_k(t+1)}{p_k(t)}\big) + \sum_{i\in A} c_i\cdot p_i(t)\cdot\log\big(\frac{p_i(t)}{w_i(t+1)}\big)\right) \\
&= \frac{1}{\eta}\Bigg(\sum_{i\in S} c_i\cdot q_i\cdot\log\big(\frac{w_k(t+1)}{p_k(t+1)}\big) + \sum_{i\in S} c_i\cdot q_i\cdot\log\big(\frac{p_k(t+1)}{p_k(t)}\big) \\
&\qquad + \sum_{i\in A} c_i\cdot p_i(t)\cdot\log\big(\frac{p_i(t)}{w_i(t+1)}\big)\Bigg)
\end{aligned} \tag{32}$$

Now, note that due to the normalization step of Algorithm 1, for any $i \in S$ we have:

$$c_i\cdot\log(\frac{w_i(t+1)}{p_i(t+1)}) = \lambda = \sum_{j\in A} c_j\cdot p_j(t+1)\cdot\frac{\lambda}{c_j} = \sum_{j\in A} c_j\cdot p_j(t+1)\cdot\log(\frac{w_j(t+1)}{p_j(t+1)})$$

So the first summation in (32) is equal to:

$$\begin{aligned}
\sum_{i\in S} c_i\cdot q_i\cdot\log\big(\frac{w_k(t+1)}{p_k(t+1)}\big) &= \sum_{i\in S} q_i\cdot\sum_{j\in A} c_j\cdot p_j(t+1)\cdot\log(\frac{w_j(t+1)}{p_j(t+1)}) \\
&= \sum_{j\in A} c_j\cdot p_j(t+1)\cdot\log(\frac{w_j(t+1)}{p_j(t+1)}) \\
&= \sum_{i\in A} c_i\cdot p_i(t+1)\cdot\log(\frac{w_i(t+1)}{p_i(t+1)})
\end{aligned} \tag{33}$$

Combining Eqn. (32) and (33), we have:

$$\begin{aligned}
\mathbf{g}(t)\cdot\big(\mathbf{q}-\mathbf{p}(t)\big) = \frac{1}{\eta}\sum_{i\in A} c_i\cdot\left(p_i(t)\cdot\log(\frac{p_i(t)}{w_i(t+1)}) + p_i(t+1)\cdot\log(\frac{w_i(t+1)}{p_i(t+1)})\right) \\
+ \frac{1}{\eta}\sum_{i\in S} c_i\cdot q_i\cdot\log\big(\frac{p_i(t+1)}{p_i(t)}\big)
\end{aligned}$$

The 2nd part is a telescopic sum when we sum over t. We will upper bound the 1st part as follows. By $\log(x) \le (x-1)$, we get that:

$$\sum_{i\in A} c_i \cdot \left(p_i(t) \cdot \log(\frac{p_i(t)}{w_i(t+1)}) + p_i(t+1) \cdot \log(\frac{w_i(t+1)}{p_i(t+1)}) \right)$$
$$\le \sum_{i\in A} c_i \cdot \left(p_i(t) \cdot \log(\frac{p_i(t)}{w_i(t+1)}) - p_i(t+1) + w_i(t+1) \right)$$
$$= \sum_{i\in A} c_i \cdot \big(p_i(t) - p_i(t+1)\big) + \sum_{i\in A} c_i \cdot \left(p_i(t) \cdot \log(\frac{p_i(t)}{w_i(t+1)}) - p_i(t) + w_i(t+1) \right)$$

Again, the 1st part is a telescopic sum when we sum over t. We will further work on the 2nd part. By the relation between $w_i(t+1)$ and $p_i(t)$, we get that:

$$\sum_{i\in A} c_i \cdot \left(p_i(t) \cdot \log(\frac{p_i(t)}{w_i(t+1)}) - p_i(t) + w_i(t+1) \right) = \sum_{i\in A} c_i \cdot p_i(t) \left(-\eta \cdot \frac{g_i(t)}{c_i} - 1 + \exp(\eta \cdot \frac{g_i(t)}{c_i}) \right)$$

Note that $\eta \cdot \frac{g_i(t)}{c_i} \in [-1, 1]$ because $g_i(t) \in [-c_i, c_i]$ and $0 < \eta \le 1$. By $\exp(x) - x - 1 \le x^2$ for $-1 \le x \le 1$ and that $\eta g_i(t) \in [-c_i, c_i]$, the above is upper bounded by $\eta^2 \sum_{i\in A} p_i(t) \frac{(g_i(t))^2}{c_i}$. Putting together, we get that:

$$\mathbf{g}(t) \cdot \big(\mathbf{q} - \mathbf{p}(t)\big) \le \frac{1}{\eta} \sum_{i\in S} c_i \cdot \left(q_i \cdot \log\big(\frac{p_i(t+1)}{p_i(t)}\big) + p_i(t) - p_i(t+1) \right) + \eta \sum_{i\in A} p_i(t) \frac{(g_i(t))^2}{c_i}$$

Summing over t, we have:

$$\mathbf{g}(t) \cdot \big(\mathbf{q} - \mathbf{p}(t)\big) \le \frac{1}{\eta} \sum_{i\in S} c_i \cdot \left(q_i \cdot \log\big(\frac{p_i(T+1)}{p_i(1)}\big) + p_i(1) - p_i(T+1) \right) + \eta \sum_{t\in[T]} \sum_{i\in A} p_i(t) \frac{(g_i(t))^2}{c_i}$$

Finally, by $\log(x) \le (x-1)$, we get that $q_i \log\big(\frac{p_i(T+1)}{q_i}\big) \le p_i(T+1) - q_i$. Hence, we have:

$$\mathbf{g}(t) \cdot \big(\mathbf{q} - \mathbf{p}(t)\big) \le \frac{1}{\eta} \sum_{i\in S} c_i \cdot \left(q_i \cdot \log\big(\frac{q_i}{p_i(1)}\big) + p_i(1) - q_i \right) + \eta \sum_{t\in[T]} \sum_{i\in A} p_i(t) \frac{(g_i(t))^2}{c_i}$$

The lemma then follows by our choice of the initial distribution. ■

A.3 Proof of OMD regret bound

In order to prove the OMD regret bound, we need some properties of Bregman divergence.

Lemma 27 (Properties of Bregman divergence (Bubeck, 2011)) *Suppose $F(\cdot)$ is a Legendre function and $D_F(\cdot,\cdot)$ is its associated Bregman divergence as defined in Definition 4. Then:*

- *$D_F(x, y) > 0$ if $x \neq y$ as F is strictly convex, and $D_F(x, x) = 0$.*

- *$D_F(., y)$ is a convex function for any choice of y.*
- *(*Pythagorean theorem*) If $\mathcal{A}$ is a convex set, $a \in \mathcal{A}$, $b \notin \mathcal{A}$ and $c = \underset{x \in \mathcal{A}}{argmin}\,(D_F(x, b))$, then*

$$D_F(a, c) + D_F(c, b) \le D_F(a, b)$$

Given Lemma 27, we are now ready to prove Lemma 6.

Proof [Proof of Lemma 6] To obtain the OMD regret bound, we have:

$$\begin{aligned}\mathbf{q} \cdot \mathbf{g}(t) - \mathbf{p}(t) \cdot \mathbf{g}(t) &= \frac{1}{\eta}(\mathbf{q} - \mathbf{p}(t)) \cdot (\nabla F(\mathbf{w}(t+1)) - \nabla F(\mathbf{p}(t))) \\ &= \frac{1}{\eta}(D_F(qb, \mathbf{p}(t)) + D_F(\mathbf{p}(t), \mathbf{w}(t+1)) - D_F(qb, \mathbf{w}(t+1))) \\ &\overset{(1)}{\le} \frac{1}{\eta} D_F(\mathbf{p}(t), \mathbf{w}(t+1)) + \frac{1}{\eta}\left(D_F(\mathbf{q}, \mathbf{p}(t)) - D_F(\mathbf{q}, \mathbf{p}(t+1))\right) \qquad (34)\end{aligned}$$

where in (1) we use $D_F(\mathbf{p}(t+1), \mathbf{w}(t+1)) \ge 0$ and $D_F(\mathbf{q}, \mathbf{p}(t+1)) + D_F(\mathbf{p}(t+1), \mathbf{w}(t+1)) \le D_F(\mathbf{q}, \mathbf{w}(t+1))$ due to Pythagorean theorem (Lemma 27). By summing up both hand sides of (34) for $t = 1, \cdots, T$ we have:

$$\sum_{t \in [T]} \mathbf{g}(t) \cdot \big(\mathbf{q} - \mathbf{p}(t)\big) \le \frac{1}{\eta} \sum_{t \in [T]} D_F(\mathbf{p}(t), \mathbf{w}(t+1)) + \frac{1}{\eta} D_F(\mathbf{q}, \mathbf{p}(1)) \qquad (35)$$

■