Zeroth-Order Non-Convex Learning via Hierarchical Dual Averaging

Amélie Héliou[1] Matthieu Martin[1] Panayotis Mertikopoulos[2 1] Thibaud Rahier[1]

Abstract

We propose a hierarchical version of dual averaging for zeroth-order online non-convex optimization – i.e., learning processes where, at each stage, the optimizer is facing an unknown non-convex loss function and only receives the incurred loss as feedback. The proposed class of policies relies on the construction of an online model that aggregates loss information as it arrives, and it consists of two principal components: (*a*) a regularizer adapted to the *Fisher information metric* (as opposed to the metric norm of the ambient space); and (*b*) a principled exploration of the problem's state space based on an adapted hierarchical schedule. This construction enables sharper control of the model's bias and variance, and allows us to derive tight bounds for both the learner's static and dynamic regret – i.e., the regret incurred against the best dynamic policy in hindsight over the horizon of play.

1. Introduction

Zeroth-order – or *derivative-free* – optimization concerns the problem of optimizing a given function without access to its gradient, stochastic or otherwise. Its study dates back at least to Rosenbrock (1960), and it has recently attracted significant interest in machine learning and artificial intelligence due to the prohibitive cost of automatic differentiation in very large neural nets and language models.

A popular approach to zeroth-order optimization involves sampling the function to be optimized at several nearby points, using the observed values to reconstruct the gradient of the function, and then employing a standard, first-order method (Conn et al., 2009). This approach allows the optimizer to approximate the gradient of the function to arbitrary precision (at least, if enough queries are made). However, it also requires that the problem's objective remain stationary during the query process.

Motivated by applications to online ad auctions and recommender systems, our paper concerns the case where this stationarity assumption breaks down – the *zeroth-order online optimization* (ZOO) setting. Specifically, we consider an adversarial ZOO problem that unfolds as follows:

1. At each stage $t = 1, 2, \dots$, the optimizer selects an action x_t from a compact convex subset $\mathcal{K}$ of $\mathbb{R}^d$.
2. Simultaneously, an adversary selects a reward function $u_t \colon \mathcal{K} \to \mathbb{R}$, often assumed to take values in $[0, 1]$.
3. The optimizer receives $u_t(x_t)$ as a reward, and the process repeats.

The learner's performance after T stages is measured here by their regret, viz. $R_T = \sum_{t=1}^{T} [u_t(x) - u_t(x_t)]$, and the learner's goal is to minimize the growth rate of R_T.

Since each individual u_t may be encountered once – and only once – it is no longer possible to perform multiple queries per function. On that account, the simultaneous perturbation stochastic approximation (SPSA) estimator of Spall (1992) has been studied extensively as a viable alternative to multiple-point query methods for online optimization. In particular, using a variant of the SPSA scheme, Flaxman et al. (2005) showed that it is possible to achieve $\mathcal{O}(T^{3/4})$ regret if the payoff functions encountered are concave. The corresponding lower bound is $\Omega(T^{1/2})$, and it was only recently achieved by the kernel-based method of Bubeck & Eldan (2016) and Bubeck et al. (2017).

When venturing beyond problems with a convex structure, the situation is significantly more complicated. The most widely studied case is the "Lipschitz bandit" – or, sometimes, "Hölder bandit" – framework where each u_t is a random realization of a parametric model of the form $u_t(x) = \hat{u}(x; \xi_t)$ with Lipschitz continuous mean $u(x) = \mathbb{E}_\xi[\hat{u}(x; \xi)]$, cf. Agrawal (1995). In this case, the lower bound for the regret is $\Omega(T^{\frac{d+1}{d+2}})$, and several algorithms have been proposed to achieve it, typically by combining an intelligent discretization of the problem's search region with a deterministic UCB-type policy (Bubeck et al., 2011; Kleinberg et al., 2008; Slivkins, 2019).

On the other hand, in an adversarial setting, an informed

Authors appear in alphabetical order. [1]Criteo AI Lab [2]Univ. Grenoble Alpes, CNRS, Inria, Grenoble INP, LIG, 38000 Grenoble, France. Correspondence to: Panayotis Mertikopoulos <panayotis.mertikopoulos@imag.fr>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

adversary can always impose $\Omega(T)$ regret to any *deterministic* decision algorithm employed by the learner, cf. Hazan et al. (2017); Shalev-Shwartz (2011); Suggala & Netrapalli (2020). This makes the algorithms designed for Lipschitz bandits ill-suited for the framework at hand, and necessitates a different approach. In this direction, Krichene et al. (2015) showed that, if each payoff function u_t is revealed to the learner after playing, it is possible to achieve $\mathcal{O}(T^{1/2})$ regret. Similar bounds were obtained more recently by Agarwal et al. (2019) and Suggala & Netrapalli (2020), who examined the *"follow the perturbed leader"* (FTPL) algorithm of Kalai & Vempala (2005) assuming access to an offline optimization oracle; however, the knowledge of u_t is still implicitly required in these works (as input to an optimization or sampling oracle, depending on the context).

More recently, Héliou et al. (2020) proposed a general dual averaging framework for online non-convex learning with imperfect feedback, including the bona fide, adversarial ZOO case. Specifically, by using a "kernel smoothing" method in the spirit of Bubeck et al. (2017), Héliou et al. (2020) proposed a ZOO method achieving *a*) a sub-optimal $\mathcal{O}(T^{\frac{d+2}{d+3}})$ regret bound; and *b*) a commensurate $\mathcal{O}(T^{\frac{d+3}{d+4}} V_T^{\frac{1}{d+4}})$ bound for the learner's *dynamic* regret, with $V_T = \sum_{t=1}^{T} \|u_{t+1} - u_t\|_\infty$ denoting the *total variation* of the payoff functions encountered (a common dynamic regret benchmark introduced by Besbes et al., 2015). However, the kernel method employed by Héliou et al. (2020) is difficult to implement because the kernel's support function may grow exponentially in both T and d.

Our contributions. In this paper, we take a different approach that fuses the dual averaging framework of Krichene et al. (2015) with a hierarchical exploration scheme in the spirit of Bubeck et al. (2011) and Kleinberg et al. (2008; 2019). Specifically, we propose a flexible, anytime *hierarchical dual averaging* (HDA) method with the following desirable properties: *(i)* it enjoys a min-max optimal $\mathcal{O}(T^{\frac{d+1}{d+2}})$ static regret bound; *(ii)* it guarantees at most $\mathcal{O}(T^{\frac{d+2}{d+3}} V_T^{\frac{1}{d+3}})$ dynamic regret. In this way, our paper closes the optimality gap in the regret analysis of Héliou et al. (2020), and it answers in the positive the authors' conjecture that it is possible to achieve $\mathcal{O}(T^{\frac{d+2}{d+3}} V_T^{\frac{1}{d+3}})$ dynamic regret in adversarial ZOO problems.

As far as we are aware, HDA is the first algorithm in the literature enjoying this dynamic regret guarantee. Moreover, in contrast to the CAB algorithm of Kleinberg (2004), we should stress that HDA *does not* require a restart schedule or a doubling trick. From a practical viewpoint, this is particularly important because the doubling trick leads to sharp performance drops when the algorithm periodically restarts from scratch – an unpleasant property, which is one of the main reasons that doubling methods are rarely employed by practitioners (Bubeck et al., 2011).

Our analysis relies on two principal components: *a*) a logarithmic scheduler for controlling the hierarchical exploration of the problem's state space; and *b*) a regularization framework adapted to the Fisher information metric on the learner's mixed strategies. The first of these components marks a crucial point of departure from the hierarchical approach of Bubeck et al. (2011) and Kleinberg et al. (2019) since, instead of increasing the granularity of our search "pointwise", we do so "dimension-wise" (but at a slower pace). As for the second component, the use of the Fisher information metric allows us to drop the reliance of dual averaging on a global norm that is not adapted to the geometry of the problem at hand, and it allows us to bring into play a wide range of regularizers that were previously unexplored in the literature – such as the Burg entropy. This is a crucial difference with existing results on dual averaging, and it allows for much finer control of the learning process as it unfolds – precisely because the information content of the learner's policy is not ignored in the process.

Upon completion of our paper, we discovered a very recent preprint by Podimata & Slivkins (2021) that proposes an adversarial zooming algorithm. The authors achieve a static $\mathcal{O}(T^{\frac{d+1}{d+2}})$ regret bound in high probability (but do not provide any dynamic regret guarantees). Their algorithm uses an explicit exploration term, plus a confidence term in the per-round sampling uncertainty. Their splitting rule splits only one-by-one cover set into 2^d sub-covers, which might be more difficult to implement in practice.

2. Setup and preliminaries

2.1. The model

We assume throughout that $\mathcal{K}$ is a compact convex subset of an ambient real space $\mathbb{R}^d$ endowed with an abstract norm $\|\cdot\|$ and a reference measure λ (typically the ordinary Lebesgue measure). As for the payoff functions encountered by the learner, we will make the following blanket assumption:

Assumption 1. The stream of payoff functions $u_t \colon \mathcal{K} \to \mathbb{R}$, $t = 1, 2, \dots$, is *uniformly bounded Lipschitz*, i.e., there exist nonnegative constants $R, L \geq 0$ such that

1. $0 \leq u_t(x) \leq R$ for all $x \in \mathcal{K}$.
2. $|u_t(x') - u_t(x)| \leq L\|x' - x\|$ for all $x, x' \in \mathcal{K}$.

To avoid exploitable, deterministic strategies, we will assume that the learner has access to an unobservable randomizer that can be used to choose an action $x \in \mathcal{K}$ by means of a probability distribution on $\mathcal{K}$ – that is, a *mixed strategy*. Of course, in complete generality, the space of all mixed strategies is impractical to work with because it contains probability distributions that cannot be described in closed form (let alone have a "sampling-friendly" structure). For this reason, we will focus on *simple strategies*, i.e., probability distributions with a piecewise constant density.

Definition 1. A mixed strategy on $\mathcal{K}$ is called *simple* if it admits a density function of the form $q = \sum_{i=1}^{m} \alpha_i \, \mathbb{1}_{\mathcal{S}_i}$ for a collection of weights $\alpha_i > 0$, $i = 1, \dots, m$, and mutually disjoint λ-measurable subsets $\mathcal{S}_i$ of $\mathcal{K}$ ($\mathcal{S}_i \cap \mathcal{S}_j = \varnothing$ for $i \neq j$) such that $\int_{\mathcal{K}} q = \sum_i \alpha_i \lambda_i(\mathcal{S}_i) = 1$. The space of simple strategies on $\mathcal{K}$ will be denoted by $\mathcal{Q}(\mathcal{K})$, and the expectation of a function $f \colon \mathcal{K} \to \mathbb{R}$ under q will be written as $\langle f, q \rangle := \mathbb{E}_{x \sim q}[f(x)] = \sum_{i=1}^{m} \alpha_i \int_{\mathcal{S}_i} f(x) \, d\lambda(x)$

Owing to their decomposable structure, simple strategies are relatively easy to sample from, and they can approximate general distributions on $\mathcal{K}$ to arbitrary precision – formally, they are dense in the weak topology of (regular) probability measures on $\mathcal{K}$ (Folland, 1999, Chap. 2). On the other hand, this "universal approximation" guarantee comes at the cost of an increased number of supporting sets $\mathcal{S}_i$, $i = 1, \dots, m$. In particular, there is no "free lunch": when m grows large, sampling from a simple strategy can become computationally expensive – if not intractable – so we will pay particular attention to the support of such strategies.

Remark 1. To facilitate sampling, we will also consider strategies of the form $q = \sum_{i=1}^{m} \alpha_i \psi_{\mathcal{S}_i}$ where $\psi_{\mathcal{S}}$ is supported on $\mathcal{S}$ and can be sampled cheaply – e.g., $\psi_{\mathcal{S}}$ could be a suitably weighted Dirac distribution on a specific point of $\mathcal{S}$. Strategies of this type are not *stricto sensu* "simple", but our results will also cover this case, cf. Section 4.

2.2. Regret: static and dynamic

Going back to the learner's sequence of play, we will assume that, at each stage $t = 1, 2, \dots$, the learner picks an action $x_t \in \mathcal{K}$ based on a simple strategy $q_t \in \mathcal{Q}$, and receives the reward $u_t(x_t)$. The *regret* of the policy q_t against a *benchmark action* $x \in \mathcal{K}$ is then defined as the difference between the player's mean cumulative payoff under q_t and x over a horizon of T rounds. Formally, we have

$$\operatorname{Reg}_x(T) := \sum\nolimits_{t=1}^{T} \mathbb{E}_{x_t \sim q_t}[u_t(x) - u_t(x_t)]. \tag{1}$$

Moreover, letting $x^* \in \arg\max_{x \in \mathcal{K}} \sum_{t=1}^{T} u_t(x)$ be the "best fixed action in hindsight" over the horizon T, we also define the learner's *static regret* as

$$\operatorname{Reg}(T) := \operatorname{Reg}_{x^*}(T) = \max\nolimits_{x \in \mathcal{K}} \operatorname{Reg}_x(T). \tag{2}$$

Finally, to relax the requirement of using a "fixed" action as a comparator, we will also consider the learner's *dynamic regret*, defined here as

$$\operatorname{DynReg}(T) := \sum\nolimits_{t=1}^{T} \max_{x \in \mathcal{K}} \mathbb{E}_{x_t \sim q_t}[u_t(x) - u_t(x_t)], \tag{3}$$

i.e., as the difference between the player's mean cumulative payoff and that of the best sequence of actions $x_t^* \in \arg\min_x u_t(x)$ over the horizon of play T. Of course, in regard to its static counterpart, the agent's dynamic regret is considerably more ambitious, and achieving sublinear dynamic regret is not always possible; we examine this issue in detail in Section 5.

In both cases, it should also be clear that there is no simple strategy that can match the exact performance of the "best" action (x^* or x_t^*, depending on the context). For example, consider the static optimization problem $u_t(x) = 1 - x^2/2$ with $x \in \mathcal{K} = [-1, 1]$: then, any simple strategy $q \in \mathcal{Q}$ would yield a payoff strictly less than 1 at each round because it is sampling with probability 1 points other than 0. Nevertheless, the following lemma shows that the propagated error on the regret can be made arbitrarily small:

Lemma 1. *Let $\mathcal{U}$ be a neighborhood of $x \in \mathcal{K}$. Then, for every simple strategy $q \in \mathcal{Q}$ supported on $\mathcal{U}$, we have*

$$\operatorname{Reg}_x(T) \leq L \operatorname{diam}(\mathcal{U}) T + \sum\nolimits_{t=1}^{T} \langle u_t, q - q_t \rangle \tag{4}$$

Proof. By Assumption 1, we have $u_t(x) \leq u_t(x') + L\|x - x'\| \leq u_t(x') + L \operatorname{diam}(\mathcal{U})$ for all $x' \in \mathcal{U}$. Hence, letting $x' \sim q$ and expectations on both sides, we get $u_t(x) \leq \langle u_t, q \rangle + L \operatorname{diam}(\mathcal{U})$. Our claim then follows by summing over t and invoking the definition of the regret. ■

Remark 2. We note here that the bound (4) does not need the full capacity of the Lipschitz continuity framework; in fact, it continues to hold under much less restrictive notions, such as the weak one-sided continuity condition of Bubeck et al. (2011). Nevertheless, in the sequel we will maintain the assumption of Lipschitz continuity for simplicity.

Remark 3. We should also state here that, in the sequel, $\mathcal{U}$ will be chosen small relative to T, so the term in (4) becomes sublinear in the analysis. In more detail, in the proof of our main regret bounds, Lemma 1 will be applied several times, over windows of different lengths, and $\mathcal{U}$ will be chosen at each window to be a progressively smaller set. The exact mechanism is detailed in Appendix C.

3. Dual averaging with an explicit cover

To build some intuition for the analysis to come, we begin by adapting the *dual averaging* (DA) algorithm of Nesterov (2009) to the (infinite) space of simple strategies with an explicit cover. This will allow us to introduce the relevant notions that we will need in the sequel, namely the *range* of an estimator and the *Fisher information metric*.

3.1. Basic setup

Let $\mathcal{P} = \{\mathcal{S}_1, \dots, \mathcal{S}_m\}$ be a measurable partition of $\mathcal{K}$ with nontrivial covering sets, i.e., $\lambda(\mathcal{S}) > 0$ and $\mathcal{S} \cap \mathcal{S}' = \varnothing$

for all $\mathcal{S}, \mathcal{S}' \in \mathcal{P}$ with $\mathcal{S} \neq \mathcal{S}'$. In particular, this implies that every point $x \in \mathcal{K}$ belongs to a unique element of $\mathcal{P}$, denoted below by $\mathcal{S}_x$. Since the elements of $\mathcal{P}$ cover $\mathcal{K}$ in an unambiguous way, we will refer to $\mathcal{P}$ as an *explicit cover* of $\mathcal{K}$. This cover will be assumed fixed throughout this section.

In terms of sampling actions from $\mathcal{K}$, the above also gives rise to a set of *simple strategies* supported on $\mathcal{P}$, namely

$$\mathcal{Q}_{\mathcal{P}} = \{\textstyle\sum_{\mathcal{S}} \alpha_{\mathcal{S}} \mathbb{1}_{\mathcal{S}} : \alpha_{\mathcal{S}} \geq 0, \sum_{\mathcal{S}} \alpha_{\mathcal{S}} \lambda(\mathcal{S}) = 1\} \tag{5}$$

Geometrically, it will be convenient to interpret $\mathcal{Q}_{\mathcal{P}}$ as a simplex embedded in the space of all test functions $\phi\colon \mathcal{K} \to \mathbb{R}$ that are piecewise constant on the covering sets of $\mathcal{P}$. Since such functions may be viewed equivalently as functions $\phi\colon \mathcal{P} \to \mathbb{R}$, we will denote this function space by $\mathbb{R}^{\mathcal{P}}$.

Moving forward, we will assume that the learner is sampling from $\mathcal{K}$ with simple strategies taken from $\mathcal{Q}_{\mathcal{P}}$, and we will write $q_{\mathcal{S}} := \mathbb{P}_{x \sim q}(x \in \mathcal{S}) = \int_{\mathcal{S}} q = \alpha_{\mathcal{S}} \lambda(\mathcal{S})$ for the probability of choosing an element of $\mathcal{S}$ under q. Accordingly, our non-convex learning framework may be encoded in more concrete terms as follows: (*i*) at each stage $t = 1, 2, \dots$, the adversary chooses (but does not reveal) a payoff function $u_t\colon \mathcal{K} \to [0, R]$; (*ii*) the learner selects an action $x_t \in \mathcal{K}$ based on some simple strategy X_t supported on $\mathcal{P}$; and (*iii*) the corresponding reward $u_t(x_t)$ is received by the learner and the process repeats.

As an algorithmic template for learning in this setting, we will consider an adaptation of the classical dual averaging algorithm of Nesterov (2009). Specifically, we will focus on an online policy that we call *dual averaging with an explicit cover* (DAX), and which is defined recursively as

$$\begin{aligned} S_{t+1} &= S_t + \hat{u}_t \\ x_{t+1} \sim X_{t+1} &= Q(\eta_{t+1} S_{t+1}) \end{aligned} \tag{DAX}$$

where

1. $\hat{u}_t \in \mathbb{R}^{\mathcal{P}}$ is an *estimate* – or *model* – of the otherwise unobserved payoff function u_t of stage t.
2. $S_t \in \mathbb{R}^{\mathcal{P}}$ is an auxiliary *scoring function* that aggregates previous payoff models – so $S_t(x)$ indicates the learner's propensity of choosing $x \in \mathcal{K}$ at stage t.
3. $\eta_t > 0$ is a "learning rate" parameter that adjusts the sharpness of the learning process.
4. $Q\colon \mathbb{R}^{\mathcal{P}} \to \mathcal{Q}_{\mathcal{P}}$ is a *choice map* that transforms scoring functions $S_t \in \mathbb{R}^{\mathcal{P}}$ into simple strategies $X_t \in \mathcal{Q}_{\mathcal{P}}$.

Each component of the method is discussed in detail below. We also note that this method is often referred to as *"follow the regularized leader"* (FTRL), cf. Shalev-Shwartz (2011); Shalev-Shwartz & Singer (2006). Our choice of terminology follows Nesterov (2009) and Xiao (2010).

3.2. The choice map

We begin by detailing the method's "choice map" $Q\colon \mathbb{R}^{\mathcal{P}} \to \mathcal{Q}_{\mathcal{P}}$ which determines action choice probabilities based on the "score function" $S_t(x)$. With this in mind, we will focus on a class of "regularized strategies" that output at each stage a simple strategy $X_t \in \mathcal{Q}_{\mathcal{P}}$ that maximizes the learner's expected score minus a regularization penalty.

Specifically, we will consider choice maps of the form

$$Q(y) = \arg\max_{q \in \mathcal{Q}_{\mathcal{P}}} \{\langle y, q\rangle - h(q)\} \quad \text{for all } y \in \mathbb{R}^{\mathcal{P}}, \tag{6}$$

where the *regularizer* $h\colon \mathcal{Q}_{\mathcal{P}} \to \mathbb{R}$ is assumed to be continuous and strictly convex on $\mathcal{Q}_{\mathcal{P}}$. To streamline our presentation, we will further assume that h is *decomposable*, i.e., it can be written as $h(q) = \sum_{\mathcal{S} \in \mathcal{P}} \theta(q_{\mathcal{S}})$ for some strictly convex, C^2-smooth function $\theta\colon (0, 1] \to \mathbb{R}$. Two widely used examples are as follows:

Example 1 (Negentropy)**.** Consider the *entropic kernel* $\theta(x) = x \log x$ with the continuity convention $0 \log 0 = 0$. Then, by a standard calculation, the associated choice map is given by the logit choice model

$$\Lambda(y) = \frac{\exp(y)}{\int_{\mathcal{K}} \exp(y)}, \tag{7}$$

where $y \equiv y(x)$ is an arbitrary piecewise constant function on $\mathcal{P}$. The entropic regularizer has a very long history in the field of (online) optimization; for a (highly incomplete) list of references, see Nemirovski & Yudin (1983), Auer et al. (1995; 2002b), Beck & Teboulle (2003), Shalev-Shwartz (2011), Bubeck & Cesa-Bianchi (2012), Arora et al. (2012), Mertikopoulos & Staudigl (2018), Kleinberg et al. (2019), Slivkins (2019), Podimata & Slivkins (2021), and references therein.

Example 2 (Log-barrier)**.** Another important example is the *log-barrier* (or *Burg entropy*) kernel $\theta(x) = -\log x$. In this case, the associated choice map does not admit a closed form expression, but it can be calculated by a binary search algorithm in logarithmic time.[1] This choice has deep links to Karmarkar's "affine scaling" method for linear programming (Karmarkar, 1990; Vanderbei et al., 1986), cf. Alvarez et al. (2004), Bauschke et al. (2017), Mertikopoulos & Sandholm (2016; 2018), Bomze et al. (2019), Antonakopoulos et al. (2019; 2021), and references therein. For a recent use of the log-barrier function in the context of stochastic and/or contextual multi-armed bandit problems, see Wei & Luo (2018), Pogodin & Lattimore (2019), and Auer et al. (2019).

[1] This is done by noting that any solution of the defining maximization problem (6) would have to satisfy the first-order optimality condition $\sum_{\mathcal{S} \in \mathcal{P}} (\xi - y_{\mathcal{S}})^{-1} = 1$ for some $\xi > \max_{\mathcal{S}} y_{\mathcal{S}}$ (in which region the function being searched is strictly decreasing).

3.3. Estimators

The second basic ingredient of (DAX) is the estimate $\hat{u}_t$ of the learner's payoff function u_t at time t. Since we are working with a fixed cover $\mathcal{P}$ of $\mathcal{K}$, the estimator $\hat{u}_t$ may not exceed the cover's granularity, which is why we require $\hat{u}_t$ to be piecewise constant on $\mathcal{P}$ – i.e., $\hat{u}_t \in \mathbb{R}^{\mathcal{P}}$.

Overall, we will measure the quality of $\hat{u}_t$ as an estimator by means of the corresponding error process $Z_t = \hat{u}_t - u_t$ which is assumed to capture all sources of uncertainty and lack of precision in the learner's estimation process. To differentiate further between random (zero-mean) and systematic (nonzero-mean) errors, we will decompose Z_t as

$$Z_t = U_t + b_t, \tag{8}$$

where $b_t = \mathbb{E}[Z_t \,|\, \mathcal{F}_t]$ denotes the *bias* of the estimator, and $U_t = Z_t - b_t$ the inherent *random noise* (so $\mathbb{E}[U_t \,|\, \mathcal{F}_t] = 0$ for all t). In terms of measurability, these processes are all conditioned on the history $\mathcal{F}_t \coloneqq \mathcal{F}(X_1, \dots, X_t)$ of the learner's policy up to – and including – stage t. Thus, in terms of the sequence of events described earlier, X_t is $\mathcal{F}_t$-measurable (by definition), but x_t, Z_t, U_t and b_t are not.

For concreteness, we provide some examples below:

Example 3 (Importance weighted estimator)**.** Motivated by the literature on multi-armed bandits (Bubeck & Cesa-Bianchi, 2012; Lattimore & Szepesvári, 2020; Slivkins, 2019), a natural way to reconstruct u_t is via the *importance weighted estimator*

$$\hat{u}_t(x) = R - \frac{R - u_t(x_t)}{X_{\mathcal{S}_t,t}} \mathbb{1}(x \in \mathcal{S}_t), \tag{IWE}$$

where $\mathcal{S}_t \coloneqq \mathcal{S}_{x_t}$ denotes the element of $\mathcal{P}$ containing the sampled action x_t, and R is one upper bound of the learner's rewards. This particular formulation of (IWE) is known as "loss-based"; other normalizations are possible but this is the most widely used one when considering sampling policies based on exponential weights algorithms (Slivkins, 2019).

Example 4 (Importance weighted estimator with explicit exploration)**.** One shortfall of (IWE) is that it requires knowledge of the upper bound R for the learner's rewards. When this is not known, a suitable alternative is to introduce an *explicit exploration* parameter $\varepsilon_t > 0$ in the learner's sampling strategy X_t. This means that the learner now chooses an action $x_t \in \mathcal{P}$ according to the perturbed strategy $\hat{X}_t = (1 - \varepsilon_t) X_t + \varepsilon_t \operatorname{unif}_{\mathcal{P}}$, where $\operatorname{unif}_{\mathcal{P}} = |\mathcal{P}|^{-1} \sum_{\mathcal{S} \in \mathcal{P}} \lambda(\mathcal{S})^{-1} \mathbb{1}_{\mathcal{S}}$ denotes the uniform distribution on $\mathcal{P}$. The *importance weighted estimator with explicit exploration* is then defined as

$$\hat{u}_t(x) = \frac{u_t(x_t)}{\hat{X}_{\mathcal{S}_t,t}} \mathbb{1}(x \in \mathcal{S}_t) \tag{IWE^3}$$

with $\mathcal{S}_t \coloneqq \mathcal{S}_{x_t}$ as above. In contrast to (IWE), the estimator (IWE^3) has bias and variance bounded respectively as $\mathbb{E}[b_t] = \mathcal{O}(\varepsilon_t)$ and $\mathbb{E}[U^2_{\mathcal{S},t}] = \mathcal{O}(1/\varepsilon_t)$, i.e., both can be controlled by tuning ε_t. This provides additional flexibility relative to (IWE), but the introduction of the explicit exploration parameter ε_t often ends up having a negative impact on the regret (Slivkins, 2019), an important disadvantage.

Other estimators have also been used in the literature, such as *implicit* exploration and its variants (Kocák et al., 2014). For posterity, we only note that the set of possible values $\mathcal{R} \coloneqq \bigcup_t \operatorname{im}(\hat{u}_t) \subseteq \mathbb{R}^{\mathcal{P}}$ attained by an estimator will play an important role in the sequel. When the estimator is understood from the context, we will refer to this image set as its *range*; in the examples above, we have:

1. For (IWE): $\mathcal{R} = (-\infty, R]^{\mathcal{P}}$.
2. For (IWE^3): $\mathcal{R} = \mathbb{R}^{\mathcal{P}}_+$.

We will return to this point in the next section.

3.4. Strong convexity and the Fisher metric

Deriving explicit regret guarantees for dual averaging methods is typically contingent on the method's regularizer being *strongly convex* (Bubeck & Cesa-Bianchi, 2012; Shalev-Shwartz, 2011). Formally, strong convexity posits that there exists some $K > 0$ such that, for all $q, q' \in \mathcal{Q}$ and all $s \in [0, 1]$, we have

$$\begin{aligned} h(sq + (1-s)q') \le{}& sh(q) + (1-s)h(q') \\ &- \frac{K}{2} s(1-s) \|q - q'\|^2 \end{aligned} \tag{9}$$

In the above, $\|\cdot\|$ denotes an arbitrary reference norm on $\mathbb{R}^{\mathcal{P}}$, usually taken to be the Euclidean norm $\|\cdot\|_2$ or the Manhattan L^1 norm $\|\cdot\|_1$. However, in our case, seeing as we are comparing *probability distributions*, an arbitrary reference norm does not seem particularly adapted to the problem at hand.

Instead, when dealing with probability distributions, it is common to measure the distance of q' relative to q via the *Fisher information metric*, which is typically used to compute the informational difference between probability distributions. In our context, the Fisher metric is defined for all $q, q' \in \mathcal{Q}_{\mathcal{P}}$ with $q \ll q'$ as

$$\|q' - q\|_q^2 = \int_{\mathcal{K}} \left[\frac{d(q' - q)}{dq}\right]^2 dq = \sum_{\mathcal{S} \in \mathcal{P}} \frac{(q'_{\mathcal{S}} - q_{\mathcal{S}})^2}{q_{\mathcal{S}}}. \tag{10}$$

We will then posit the following strong convexity requirement relative to the Fisher metric

$$\begin{aligned} h(sq + (1-s)q') \le{}& sh(q) + (1-s)h(q') \\ &- \frac{K}{2} s(1-s) \|q - q'\|_q^2 \end{aligned} \tag{11}$$

for all $q, q' \in \mathcal{Q}$ and all $s \in [0, 1]$. Since this is a non-standard requirement, we proceed with an example.

Example 5. The Burg entropy $h(x) = -\sum_{\mathcal{S}\in\mathcal{P}} \log q_{\mathcal{S}}$ is 1-strongly convex relative to the Fisher metric. Indeed, since h is smooth, the strong convexity requirement for h with $K = 1$ can be rewritten as $D_{\mathrm{IS}}(q', q) \geq \frac{1}{2}\sum_{\mathcal{S}\in\mathcal{P}}(q'_{\mathcal{S}} - q_{\mathcal{S}})^2/q_{\mathcal{S}}$ where $D_{\mathrm{IS}}(q', q) = \sum_{\mathcal{S}\in\mathcal{P}}[q'_{\mathcal{S}}/q_{\mathcal{S}} - \log(q'_{\mathcal{S}}/q_{\mathcal{S}}) - 1]$ denotes the Itakura–Saito distance on $\mathcal{Q}_{\mathcal{P}}$. Our claim then follows from Antonakopoulos et al. (2020, Ex. 4).

The key implication of Fisher strong convexity for our analysis is the following characterization:

Lemma 2. *Let $h^*(y) = \max_{q\in\mathcal{Q}_{\mathcal{P}}}\{\langle y, q\rangle - h(q)\}$ be the convex conjugate of h. The following are equivalent:*

1. *h satisfies (11).*
2. *h^* is $(1/K)$-Lipschitz smooth relative to the dual Fisher norm $\|y\|_{q,*}^2 = \sum_{\mathcal{S}\in\mathcal{P}} q_{\mathcal{S}} y_{\mathcal{S}}^2$ on $\mathbb{R}^{\mathcal{P}}$; specifically, for all $y, v \in \mathbb{R}^{\mathcal{P}}$ and $\chi = Q(y)$, we have*

$$h^*(y + v) \leq h^*(y) + \langle v, \chi\rangle + \frac{1}{2K}\|v\|_{\chi,*}^2. \tag{12}$$

Lemma 2 mirrors the well-known equivalence between strong convexity in the primal and Lipschitz smoothness in the dual (Bubeck & Cesa-Bianchi, 2012; Shalev-Shwartz, 2011). However, we must stress here that the norms in (11) are *not* global, but *strategy-dependent* – in effect, they comprise a Riemannian metric on the set of simple strategies $\mathcal{Q}_{\mathcal{P}}$. This is a crucial difference with the standard analysis of dual averaging, and it allows for much finer control of the learning process as it unfolds – precisely because the base distribution $\chi = Q(y)$ is not ignored in the process.

We close this section by noting that the entropic regularizer of (1) *does not* satisfy (11); we provide an explicit discussion of this point in the supplement. However, as we also show in the supplement, it *does* satisfy the Lipschitz smoothness requirement (12) for all $v \in \mathbb{R}^{\mathcal{P}}$ that are "upper-bounded", i.e., $\sup_{\mathcal{S}\in\mathcal{P}} v_{\mathcal{S}} \leq M$ for some $M \in \mathbb{R}$. From an algorithmic viewpoint, this relaxation of (11) will play a pivotal role in the sequel, so we encode it as follows:

Definition 2. Let $\mathcal{R}$ be a nonempty convex subset of $\mathbb{R}^{\mathcal{P}}$. We say that h is K-*tame* relative to $\mathcal{R}$ if (12) holds for all $y \in \mathbb{R}^{\mathcal{P}}$ and all $v \in \mathcal{R}$.

Clearly, by Lemma 2, any regularizer satisfying (11) is tame relative to any subset of $\mathbb{R}^{\mathcal{P}}$ (including $\mathbb{R}^{\mathcal{P}}$ itself). By contrast, as we mentioned above, the entropic regularizer of Example 1 is 1-tame over the region $\mathcal{R} = \{y \in \mathbb{R}^{\mathcal{P}} : y_{\mathcal{S}} \leq 1\}$, but *it is not tame* over all of $\mathbb{R}^{\mathcal{P}}$. In the analysis to come, we will see that this property introduces an intricate interplay between the two principal components of (DAX), namely the choice of regularizer h and the estimator $\hat{u}$. Unless explicitly mentioned otherwise, in the rest of this section we will assume that $\mathcal{R}$ is fixed and h is K-tame relative to $\mathcal{R}$.

3.5. Regret analysis

The key element in our analysis will be to control the "divergence" between a scoring function S_t and a comparator strategy $q \in \mathcal{Q}_{\mathcal{P}}$. Because these two elements live in different spaces, we introduce below the *Fenchel coupling*

$$F(q, y) = h(q) + h^*(y) - \langle y, q\rangle, \tag{13}$$

for all $q \in \mathcal{Q}_{\mathcal{P}}$, $y \in \mathbb{R}^{\mathcal{P}}$. Clearly, by the Fenchel-Young inequality, we have $F(q, y) \geq 0$ with equality if and only if $Q(y) = q$. More to the point, as we show in the supplement, the Fenchel coupling enjoys the following growth property:

Lemma 3. *For all $y \in \mathbb{R}^{\mathcal{P}}$ and all $v \in \mathcal{R}$, we have*

$$\begin{aligned} F(q, y + v) &= F(q, y) + \langle v, \chi - q\rangle + F(\chi, y + v) && \text{(14a)}\\ &\leq F(q, y) + \langle v, \chi - q\rangle + \frac{1}{2K}\|v\|_{\chi,*}^2 && \text{(14b)} \end{aligned}$$

where $\chi = Q(y)$.

Using (14), we will analyze the regret properties of (DAX) via the η_t*-deflated coupling*

$$E_t = \frac{1}{\eta_t} F(q, \eta_t S_t). \tag{15}$$

Doing so leads to the following result:

Lemma 4. *Suppose that (DAX) is run with an estimator with range $\mathcal{R}$. For all $t = 1, 2, \dots$, we have*

$$\begin{aligned} E_{t+1} \leq E_t + \langle \hat{u}_t, X_t - q\rangle + \big(\eta_{t+1}^{-1} - \eta_t^{-1}\big)[h(q) - \min h]\\ + \eta_t^{-1} F(X_t, \eta_t S_{t+1}). \end{aligned} \tag{16}$$

If, in addition, h is K-tame relative to $\mathcal{R}$, the last term in (16) is bounded as

$$\eta_t^{-1} F(X_t, \eta_t S_{t+1}) \leq \eta_t/(2K)\|\hat{u}_t\|_t^2, \tag{17}$$

where $\|\cdot\|_t$ is the dual Fisher norm $\|v\|_t \coloneqq \|v\|_{X_t,*}$.

Thus, telescoping Lemma 4, we obtain the bound below.

Proposition 1. *The regret incurred relative to $q \in \mathcal{Q}_{\mathcal{P}}$ over the interval $\mathcal{T} = \{t_1, \dots, t_2 - 1\}$ is bounded as*

$$\begin{aligned} \mathrm{Reg}_q(\mathcal{T}) \leq E_{t_1} - E_{t_2} + \big(\eta_{t_2}^{-1} - \eta_{t_1}^{-1}\big)[h(q) - \min h]\\ + \sum_{t\in\mathcal{T}}\langle Z_t, X_t - q\rangle + \frac{1}{2K}\sum_{t\in\mathcal{T}}\eta_t\|\hat{u}_t\|_t^2. \end{aligned} \tag{18}$$

We are finally in a position to state our main regret guarantees for (DAX). For generality, we state our result with a generic estimator $\hat{u}_t$ enjoying the following bounds:

a) *Bias:* $\quad |\langle b_t, q\rangle| \leq \mu_t \quad$ (19a)

b) *Mean square:* $\quad \mathbb{E}[\|\hat{u}_t\|_t^2 \,|\, \mathcal{F}_t] \leq M_t^2 \quad$ (19b) for all $t = 1, 2, \dots$, and all $q \in \mathcal{Q}_{\mathcal{P}}$. We stress here that the use of the Fisher metric in (19) is *crucial*: for example, the IWE estimator satisfies (19b) with $M_t = \mathcal{O}(R^2|\mathcal{P}|)$ (where $|\mathcal{P}|$ is the size of the underlying partition) but it does not satisfy this bound for *any* global norm. Again, the reason for this is that the dual Finsler norm can be considerably smaller than any other global norm, depending on the information content of X_t.

This feature plays a key role in deriving the regret of (DAX):

Theorem 1. *Suppose that* (DAX) *is run with assumptions as in Proposition 1. Then the learner's regret is bounded as*

$$\mathbb{E}[\mathrm{Reg}_q(T)] \leq E_1 - E_{T+1} + \left(\eta_{T+1}^{-1} - \eta_1^{-1}\right)[h(q) - \min h] + 2\sum\nolimits_{t=1}^{T} \mu_t + \frac{1}{2K}\sum\nolimits_{t=1}^{T} \eta_t M_t^2. \tag{20}$$

This theorem is proved in the supplement and constitutes the main ingredient for the analysis to come.

4. Hierarchical dual averaging

In this section, we proceed to define the mechanism that we will use to recursively "zoom-in" on different regions of the state space. This hierarchical approach is inspired by earlier works by Bubeck et al. (2011), but with the crucial difference that we do not zoom in "pointwise" but "dimension-wise". We explain all this in detail below.

4.1. The splitting mechanism

As in the case of Bubeck et al. (2011) and Kleinberg et al. (2008; 2019), the basic element of our construction is an infinite "tree of coverings", each of whose levels $\sigma = 1, \dots$ defines a successively finer cover $\mathcal{P}_\sigma$ of $\mathcal{K}$ (i.e., $\mathcal{P}_\sigma \subseteq \mathcal{P}_{\sigma+1}$ for all $\sigma = 1, \dots$). However, in contrast to these previous works, we do not consider *binary* trees, but *dyadic* ones; specifically, each cover $\mathcal{P}_\sigma = \{\mathcal{S}_{\sigma,i}\}_{i \leq 2^\sigma}$ is defined inductively as follows: (*i*) $\mathcal{P}_0 = \{\mathcal{S}_{0,1}\} = \{\mathcal{K}\}$; (*ii*) at specific stages of the learning process (that we define later), a *splitting event* occurs, and each leaf[2] of the current cover is split into 2 sub-leaves as detailed below (refer also to Figs. 1 and 2 for intuition in the case $d = 2$). We perform splitting events successively along each dimension in a round-robin manner, ensuring each node is split into two subnodes of equal volume. Formally, for a given node $\mathcal{S}_{\sigma,i}$, we define $\mathcal{S}_{\sigma+1,2i-1}$ and $\mathcal{S}_{\sigma+1,2i}$ as the two subsets obtained from splitting the leaf $\mathcal{S}_{\sigma,i}$ in 2 equally sized leaves using a hyperplane[3] orthogonal to the canonical basis vector of $\mathbb{R}^d$ number $\sigma + 1 \pmod{d}$. We then have $\mathcal{S}_{\sigma+1,2i} \cup \mathcal{S}_{\sigma+1,2i-1} = \mathcal{S}_{\sigma,i}$, $\mathcal{S}_{\sigma+1,2i} \cap \mathcal{S}_{\sigma+1,2i-1} = \varnothing$ and $\lambda(\mathcal{S}_{\sigma+1,2i}) = \lambda(\mathcal{S}_{\sigma+1,2i-1}) = \lambda(\mathcal{S}_{\sigma,i})/2$.

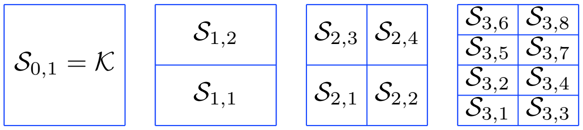

Figure 1. Example of the 3 first *splitting events* for $\mathcal{K} = [0, 1]^2$

In the sequel, for any cover $\mathcal{P}$, we write $\mathcal{P}^+$ for its *successor* cover, i.e., the cover after a splitting event on $\mathcal{P}$.

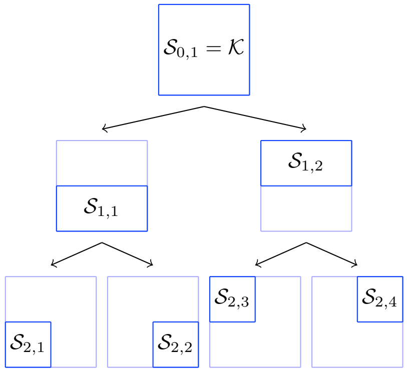

Figure 2. Example of a covering tree for the cube $\mathcal{K} = [0, 1]^2$

A crucial information for the sequel is the diameter of the leaves $\mathcal{S}_{\sigma,i}$ of a given cover $\mathcal{P}$, for which we make a geometric assumption similar to Bubeck et al. (2011, A1):

Assumption 2. There exists some $C_{\mathcal{K}} > 0$ such that

$$\mathrm{diam}(\mathcal{S}_{\sigma,i}) \leq C_{\mathcal{K}} \, \mathrm{diam}(\mathcal{K}) 2^{-\lfloor \sigma/d \rfloor}. \tag{21}$$

for all $\sigma \geq 0$ and $i \leq 2^\sigma$ as above.

Assumption 2 only concerns the problem's domain $\mathcal{K}$, and it can be lifted by embedding $\mathcal{K}$ in a suitable box and then proceeding with a splitting schedule that follows a fixed volumetric mesh. This approach could lead to leaves of different volume at each splitting event, which would in turn make the analysis more cumbersome. The example below shows that $C_{\mathcal{K}}$ can be easy to calculate in many cases:

Example 6 ($\mathcal{K}$ = d-dimensional box)**.** In the particular case where $\mathcal{K}$ is an hyperrectangle with sides parallel to the canonical basis vectors of $\mathbb{R}^d$, we have $\mathrm{diam}(\mathcal{S}_{\sigma,i}) \leq \mathrm{diam}(\mathcal{K})2^{-\lfloor \sigma/d \rfloor}$ and $C_{\mathcal{K}} = 1$.

4.2. The hierarchical dual averaging algorithm

As a prelude to the definition of our algorithm, we introduce the following notions: for all $t = 1, 2, \dots$, (*i*) $\mathcal{P}_t$ will denote the current cover at time t; (*ii*) we will write σ_t for the number of splitting events made prior to time t (so σ_t is also

[2] In a slight overload, we also write $\mathcal{P}$ for the tree inducing the cover, and therefore refer to its components as *leaves*

[3] Given a set $\mathcal{S} \subseteq \mathbb{R}^d$ and a dimension $k \in \{1, \dots, d\}$, an hyperplane with normal vector e_k which splits $\mathcal{S}$ into two equally sized subsets exists by the intermediate value theorem and can be found efficiently by line search.

the height of the tree $\mathcal{P}_t$); and (*iii*) $m_t = 2^{\sigma_t}$ will denote the number of leaves of $\mathcal{P}_t$. Moreover, a *splitting schedule* is an increasing sequence of integers $\mathcal{T}_{\text{split}} = \{t_1, t_2, \dots\}$ such that we perform a splitting event at each round $t \in \mathcal{T}_{\text{split}}$. For convenience, we will rather manipulate *scheduler sequences* $\{v_t\}_{t\geq 1}$, i.e., increasing real sequences that are uniquely mapped to a splitting schedule by $\mathcal{T}_{\text{split}}(v) = \{t \geq 1 \text{ such that } \lfloor v_t \rfloor = \lfloor v_{t-1} \rfloor + 1\}$. In the sequel and when the context is non ambiguous we may use the term *splitting schedule* to refer to its associated *scheduler sequence*. We note that for all t, these definitions imply $\lfloor v_t \rfloor = \sigma_t$, and that in the light of the relation stated in the previous subsection we have, for any $\mathcal{S} \in \mathcal{P}_t$, $\operatorname{diam}(\mathcal{S}) \leq 2 \operatorname{diam}(\mathcal{K}) m_t^{-1/d}$ and $\lambda(\mathcal{S}) = m_t^{-1}\lambda(\mathcal{K})$.

We are now in a position to define our learning algorithm in detail. Its components are threefold: (*i*) a sequence of estimators $\hat{u}_t$ with range $\mathcal{R}$; (*ii*) a regularizer that is K-tame relative to $\mathcal{R}$; and (*iii*) a splitting schedule $\mathcal{T}_{\text{split}}(v)$ as above. Then, the *hierarchical dual averaging* (HDA) is defined as

$$\begin{aligned} S_{t+1} &\leftarrow S_t + \hat{u}_t \\ x_{t+1} \sim X_{t+1} &\leftarrow Q^{\mathcal{P}_t}(\eta_{t+1} S_{t+1}) \\ \mathcal{P}_{t+1} &\leftarrow \mathcal{P}_t^{+} \text{if } t \in \mathcal{T}_{\text{split}}(v) \end{aligned} \tag{HDA}$$

where $Q^{\mathcal{P}}$ denotes the choice map induced by h for a given cover $\mathcal{P}$ of $\mathcal{K}$ (by convention, we take $\mathcal{P}_0 = \{\mathcal{K}\}$), η_t is a variable learning rate sequence, and we implicitly treat $\hat{u}_t$ and S_t as elements $\mathbb{R}^{\mathcal{P}_t}$ and X_t as an element of $\mathcal{Q}_{\mathcal{P}_t}$.

By construction, (HDA) comprises a succession of applications of (DAX) to sequences of successive rounds during which the underlying partition $\mathcal{P}_t$ stays the same (i.e., in between two successive splitting events). An important special case is the specific instance of (HDA) obtained by the entropic kernel $\theta(x) = x \log x$ (cf. Example 1) and the estimator (IWE); we will refer to this instance as the *hierarchical exponential weights* (HEW) algorithm.

5. Analysis and results

In this section, we leverage the regret guarantees established in Section 3 for (DAX) to derive a template regret bound for (HDA) – and, in particular, for HEW.

Static regret. Our template bound for HDA is as follows.

Theorem 2. *The HDA algorithm enjoys the regret bound*

$$\begin{aligned} \mathbb{E}[\operatorname{Reg}_x(T)] \leq \frac{\phi(m_T) + C_\theta \log_2(m_T)}{\eta_{T+1}} \\ + 2LC_{\mathcal{K}} \operatorname{diam}(\mathcal{K}) \sum\nolimits_{t=1}^{T} m_t^{-1/d} \\ + 2\sum\nolimits_{t=1}^{T} \mu_t + \frac{1}{2K}\sum\nolimits_{t=1}^{T} \eta_t M_t^2, \end{aligned} \tag{22}$$

where m_t is the number of sets in the partition $\mathcal{P}_t$, $\phi(z) = z\theta(1/z)$ for all $z > 0$ and C_θ is a constant depending only on θ. In particular, if (HDA) *is run with learning rate $\eta_t \propto 1/t^{\varrho}$, $\varrho \in (0,1)$, a logarithmic splitting schedule $v_t = p\log_2(t)$ and a sequence of estimators $\hat{u}_t$ such that $\mu_t = \mathcal{O}(1/t^{\beta})$ and $M_t^2 = \mathcal{O}(t^{2\mu})$ for some $\beta, \mu \geq 0$, then*

$$\mathbb{E}[\operatorname{Reg}(T)] = \mathcal{O}(\phi(T^{-p})T^{\varrho} + T^{1-p/d} + T^{1-\beta} + T^{1+2\mu-\varrho}). \tag{23}$$

The proof of Theorem 2 is presented in detail in Appendix C and hinges on applying Proposition 1 to bound the regret of (HDA) on each time window during which the algorithm maintains a constant cover of $\mathcal{K}$. Aggregating these bounds provides a regret guarantee for (HDA) over the entire horizon time of play; however, since (HDA) is *not* restarted at each window, joining the resulting window-by-window bounds ends up being fairly delicate. The main dificulties (and associated error terms in the regret) are as follows:

1. A comparator for a given time frame may not be admissible for a previous time frame because the granularity of an antecedent cover may not suffice to include the comparator in question. This propagates a “resolution error” that becomes smaller when the cover gets finer, but larger when the window gets longer.
2. At every splitting event, the algorithm retains the same probability distribution over $\mathcal{K}$ (to avoid restart-forget effects). However, this introduces a “splitting residue” because of the necessary correction in the learner’s scores when the resolution of the cover increases.

The above steps are made precise in Appendix C, where we show how each of these contributing factors can be bounded in an efficient manner. For the moment, we only note that the template bound of Theorem 2 can be used to derive tight regret bounds for particular instances of (*i*) the estimator sequence $\{\hat{u}_t\}_t$ with range $\mathcal{R}$; (*ii*) the regularizer h which is K*-tame* relative to $\mathcal{R}$; and (*iii*) the splitting schedule $\{v_t\}_t$.

We carry all this out for the HEW algorithm below:

Corollary 1. *If HEW is run with learning rate $\eta_t \propto t^{-\varrho}$ and the logarithmic splitting schedule $v_t = p\log_2(t)$, the learner enjoys the bound*

$$\mathbb{E}[\operatorname{Reg}(T)] = \mathcal{O}(T^{\varrho} + T^{1-p/d} + T^{1+p-\varrho}). \tag{24}$$

In particular, if the algorithm is run with $\varrho = (d+1)/(d+2)$ and $p = d/(d+2)$ we obtain the bound

$$\mathbb{E}[\operatorname{Reg}(T)] = \mathcal{O}(T^{\frac{d+1}{d+2}}). \tag{25}$$

Dynamic regret guarantees. We now show guarantees of HDA in terms of the dynamic regret introduced in (3). We would like to stress that the expected dynamic regret of an algorithm cannot be bounded without any restriction on the sequence of payoffs (Shalev-Shwartz, 2011). For this reason, dynamic regret guarantees are often stated in terms of the *variation* of the payoff functions $\{u_t\}_t$, defined as follows (Besbes et al., 2015)

$$V_T := \sum_{t=1}^{T} \|u_{t+1} - u_t\|_\infty, \tag{26}$$

with the convention $u_{t+1} = u_t$ for $t = T$. We then have:

Theorem 3. *Suppose that* (HDA) *is run with the negentropy kernel, and assumptions as in Theorem 2. Then:*

$$\begin{aligned} &\mathbb{E}[\mathrm{DynReg}(T)] \\ &= \mathcal{O}(T^{1+2\mu-\varrho} + T^{1-\beta} + T^{1-p/d} + T^{2\varrho-2\mu} V_T). \end{aligned} \tag{27}$$

Finally, with judiciously chosen parameters, our template bound yields the following improvement over previous dynamic regret bounds in the literature:

Corollary 2. *Suppose that HEW is run with splitting schedule $v_t = p\log_2(t)$ and learning rate $\eta_t \propto 1/t^\varrho$. Then:*

$$\mathbb{E}[\mathrm{DynReg}(T)] = \mathcal{O}(T^{1+p-\varrho} + T^{1-p/d} + T^{2\varrho-p} V_T). \tag{28}$$

Hence, if $V_T = \mathcal{O}(T^\nu)$ for some $\nu < 1$, setting $\varrho = (1-\nu)(d+1)/(d+3)$ and $p = (1-\nu)d/(d+3)$ delivers

$$\mathbb{E}[\mathrm{DynReg}(T)] = \mathcal{O}(T^{(d+2)/(d+3)} V_T^{1/(d+3)}). \tag{29}$$

This result was conjectured by Héliou et al. (2020) and, as far as we are aware, this is the first time it is achieved. The main limiting factor in the kernel-based approach of Héliou et al. (2020) is that it requires the introduction of an explicit exploration term; in turn, this leads to an unavoidable extra term in the regret, and to suboptimal regret bounds. The importance of the proposed splitting mechanism is that it does not require a kernel to smooth (IWE), and the use of the Fisher information metric allows us to control the variance of (IWE) without introducing an explicit exploration error.

We suspect that the above bound is min-max optimal, but we are not aware of any lower bounds for the dynamic regret against non-convex Lipschitz losses – this is actually stated as an open problem in the paper of Besbes et al. (2015). In particular, the analysis of Besbes et al. (2015, Theorem 2) seems to suggest that, if an informed adversary can impose $\Omega(T^q)$ *static* regret, they can also impose $\Omega(T^{1/(2-q)} V_T^{(1-q)/(2-q)})$ *dynamic* regret. If this conjecture is true, this would mean that our bound is itself tight, because the static regret exponent $q = (d+1)/(d+2)$ is well known to be tight for Lipschitz losses.

Remark 4. We should also state here that (HDA) is not parameter-free, as it implicitly requires knowledge of an upper bound for V_T. As one of the reviewers pointed out, this requirement – which was stated as an open problem in the work of Besbes et al. (2015) – has been partially resolved for (stochastic) multi-armed and contextual bandits by Auer et al. (2019); we are not aware of a similar result for adversarial online non-convex optimization problems.

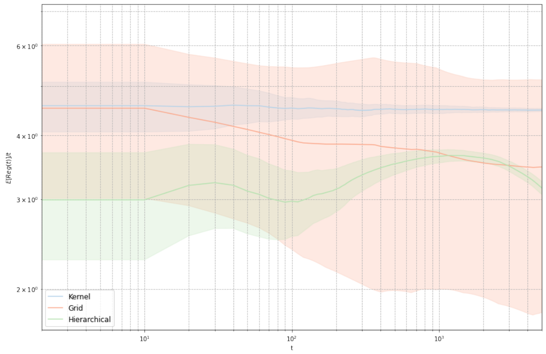

Figure 3. **Expected average regret** over 92 realizations for each algorithm (solid line). The shaded area represents one standard deviation from the mean. The three strategies compared are ”Hierarchical” (ours), ”Grid” being a naive discretized mesh on the search space, and the “Kernel” strategy from (Héliou et al., 2020)

Numerical experiments. For illustration purposes, Fig. 3 provides some numerical experiments on different no-regret policies discussed in the rest of our paper. Specifically, we compared 3 strategies, “Grid”, “Kernel” and “Hierarchical”, and plot the current instantaneous regret w.r.t. the current round t. The shaded area representing the instantaneous variance of such regret, each strategies being launched with multiple initialized seed (92). First the “Hierarchical” method is as outlined in Section 4 with parameters of the algorithm described below. Second the “Grid” method involves partitioning the search space $\mathcal{K}$ into a regular grid of a given mesh-size. This *a priori* discretization level constitute the algorithm hyperparameter. The “Grid” then treats the problem as a finite-armed bandit on the latter discretized search space, applying the EXP3 algorithm (Auer et al., 2002a). Finally, the “Kernel” strategy is based on Héliou et al. (2020), using a squared-kernel based estimate. The adversarial function is analytic and randomly drawn, with known maximum. We present the full details of our experiments in Appendix D.

Acknowledgements

P. Mertikopoulos is grateful for financial support by the French National Research Agency (ANR) in the framework of the “Investissements d’avenir” program (ANR-15-IDEX-02), the LabEx PERSYVAL (ANR-11-LABX-0025-01), MIAI@Grenoble Alpes (ANR-19-P3IA-0003), and the grants ORACLESS (ANR-16-CE33-0004) and ALIAS (ANR-19-CE48-0018-01).

References

Agarwal, N., Gonen, A., and Hazan, E. Learning in non-convex games with an optimization oracle. In *COLT '19: Proceedings of the 32nd Annual Conference on Learning Theory*, 2019.

Agrawal, R. The continuum-armed bandit problem. *SIAM Journal on Control and Optimization*, 33(6):1926–1951, November 1995.

Alvarez, F., Bolte, J., and Brahic, O. Hessian Riemannian gradient flows in convex programming. *SIAM Journal on Control and Optimization*, 43(2):477–501, 2004.

Antonakopoulos, K., Belmega, E. V., and Mertikopoulos, P. An adaptive mirror-prox algorithm for variational inequalities with singular operators. In *NeurIPS '19: Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2019.

Antonakopoulos, K., Belmega, E. V., and Mertikopoulos, P. Online and stochastic optimization beyond Lipschitz continuity: A Riemannian approach. In *ICLR '20: Proceedings of the 2020 International Conference on Learning Representations*, 2020.

Antonakopoulos, K., Belmega, E. V., and Mertikopoulos, P. Adaptive extra-gradient methods for min-max optimization and games. In *ICLR '21: Proceedings of the 2021 International Conference on Learning Representations*, 2021.

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

Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, 1995.

Auer, P., Cesa-Bianchi, N., and Fischer, P. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47: 235–256, 2002a.

Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002b.

Auer, P., Chen, Y., Gajane, P., Lee, C.-W., Luo, H., Ortner, R., and Wei, C.-Y. Achieving optimal dynamic regret for non-stationary bandits without prior information. In *COLT '19: Proceedings of the 32nd Annual Conference on Learning Theory*, 2019.

Bauschke, H. H. and Combettes, P. L. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, New York, NY, USA, 2 edition, 2017.

Bauschke, H. H., Bolte, J., and Teboulle, M. A descent lemma beyond Lipschitz gradient continuity: First-order methods revisited and applications. *Mathematics of Operations Research*, 42(2):330–348, May 2017.

Beck, A. and Teboulle, M. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

Berge, C. *Topological Spaces*. Dover, New York, 1997.

Besbes, O., Gur, Y., and Zeevi, A. Non-stationary stochastic optimization. *Operations Research*, 63(5):1227–1244, October 2015.

Bomze, I. M., Mertikopoulos, P., Schachinger, W., and Staudigl, M. Hessian barrier algorithms for linearly constrained optimization problems. *SIAM Journal on Optimization*, 29(3): 2100–2127, 2019.

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

Bubeck, S. and Eldan, R. Multi-scale exploration of convex functions and bandit convex optimization. In *COLT '16: Proceedings of the 29th Annual Conference on Learning Theory*, 2016.

Bubeck, S., Munos, R., Stoltz, G., and Szepesvári, C. $\mathcal{X}$-armed bandits. *Journal of Machine Learning Research*, 12:1655–1695, 2011.

Bubeck, S., Lee, Y. T., and Eldan, R. Kernel-based methods for bandit convex optimization. In *STOC '17: Proceedings of the 49th annual ACM SIGACT symposium on the Theory of Computing*, 2017.

Chen, G. and Teboulle, M. Convergence analysis of a proximal-like minimization algorithm using Bregman functions. *SIAM Journal on Optimization*, 3(3):538–543, August 1993.

Conn, A. R., Scheinberg, K., and Vicente, L. N. *Introduction to Derivative-Free Optimization*. Society for Industrial and Applied Mathematics, 2009.

Flaxman, A. D., Kalai, A. T., and McMahan, H. B. Online convex optimization in the bandit setting: gradient descent without a gradient. In *SODA '05: Proceedings of the 16th annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 385–394, 2005.

Folland, G. B. *Real Analysis*. Wiley-Interscience, 2 edition, 1999.

Hazan, E., Singh, K., and Zhang, C. Efficient regret minimization in non-convex games. In *ICML '17: Proceedings of the 34th International Conference on Machine Learning*, 2017.

Héliou, A., Martin, M., Mertikopoulos, P., and Rahier, T. Online non-convex optimization with imperfect feedback. In *NeurIPS '20: Proceedings of the 34th International Conference on Neural Information Processing Systems*, 2020.

Kalai, A. T. and Vempala, S. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, October 2005.

Karmarkar, N. Riemannian geometry underlying interior point methods for linear programming. In *Mathematical Developments Arising from Linear Programming*, number 114 in Contemporary Mathematics. American Mathematical Society, 1990.

Kleinberg, R. D. Nearly tight bounds for the continuum-armed bandit problem. In *NIPS' 04: Proceedings of the 18th Annual Conference on Neural Information Processing Systems*, 2004.

Kleinberg, R. D., Slivkins, A., and Upfal, E. Multi-armed bandits in metric spaces. In *STOC '08: Proceedings of the 40th annual ACM symposium on the Theory of Computing*, 2008.

Kleinberg, R. D., Slivkins, A., and Upfal, E. Bandits and experts in metric spaces. *Journal of the ACM*, 66(4), May 2019.

Kocák, T., Neu, G., Valko, M., and Munos, R. Efficient learning by implicit exploration in bandit problems with side observations. In *NIPS '14: Proceedings of the 28th International Conference on Neural Information Processing Systems*, 2014.

Krichene, W., Balandat, M., Tomlin, C., and Bayen, A. The Hedge algorithm on a continuum. In *ICML '15: Proceedings of the 32nd International Conference on Machine Learning*, 2015.

Lattimore, T. and Szepesvári, C. *Bandit Algorithms*. Cambridge University Press, Cambridge, UK, 2020.

Mertikopoulos, P. and Sandholm, W. H. Learning in games via reinforcement and regularization. *Mathematics of Operations Research*, 41(4):1297–1324, November 2016.

Mertikopoulos, P. and Sandholm, W. H. Riemannian game dynamics. *Journal of Economic Theory*, 177:315–364, September 2018.

Mertikopoulos, P. and Staudigl, M. On the convergence of gradient-like flows with noisy gradient input. *SIAM Journal on Optimization*, 28(1):163–197, January 2018.

Nemirovski, A. S. and Yudin, D. B. *Problem Complexity and Method Efficiency in Optimization*. Wiley, New York, NY, 1983.

Nesterov, Y. Primal-dual subgradient methods for convex problems. *Mathematical Programming*, 120(1):221–259, 2009.

Podimata, C. and Slivkins, A. Adaptive discretization for adversarial Lipschitz bandits. https://arxiv.org/abs/2006.12367, 2021.

Pogodin, R. and Lattimore, T. Adaptivity, variance and separation for adversarial bandits. https://arxiv.org/pdf/1903.07890.pdf, 2019.

Rockafellar, R. T. *Convex Analysis*. Princeton University Press, Princeton, NJ, 1970.

Rosenbrock, H. H. An automatic method for finding the greatest or least value of a function. *Computer Journal*, 3(3):175–184, 1960.

Shalev-Shwartz, S. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2): 107–194, 2011.

Shalev-Shwartz, S. and Singer, Y. Convex repeated games and Fenchel duality. In *NIPS' 06: Proceedings of the 19th Annual Conference on Neural Information Processing Systems*, pp. 1265–1272. MIT Press, 2006.

Slivkins, A. Introduction to multi-armed bandits. *Foundations and Trends in Machine Learning*, 12(1-2):1–286, November 2019.

Spall, J. C. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Trans. Autom. Control*, 37(3):332–341, March 1992.

Suggala, A. S. and Netrapalli, P. Online non-convex learning: Following the perturbed leader is optimal. In *ALT '20: Proceedings of the 31st International Conference on Algorithmic Learning Theory*, 2020.

Vanderbei, R. J., Meketon, M. S., and Freedman, B. A. A modification of Karmarkar's linear programming algorithm. *Algorithmica*, 1(1):395–407, November 1986.

Wei, C.-Y. and Luo, H. More adaptive algorithms for adversarial bandits. In *COLT '18: Proceedings of the 31st Annual Conference on Learning Theory*, 2018.

Xiao, L. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11:2543–2596, October 2010.