JMLR: Workshop and Conference Proceedings vol 40:1–33, 2015

A Chaining Algorithm for Online Nonparametric Regression

Pierre Gaillard PIERRE-P.GAILLARD@EDF.FR
EDF R&D, Clamart, France
GREGHEC: HEC Paris – CNRS, Jouy-en-Josas, France

Sébastien Gerchinovitz SEBASTIEN.GERCHINOVITZ@MATH.UNIV-TOULOUSE.FR
Institut de Mathématiques de Toulouse, Université Paul Sabatier, France

Abstract

We consider the problem of online nonparametric regression with arbitrary deterministic sequences. Using ideas from the chaining technique, we design an algorithm that achieves a Dudley-type regret bound similar to the one obtained in a non-constructive fashion by Rakhlin and Sridharan (2014). Our regret bound is expressed in terms of the metric entropy in the sup norm, which yields optimal guarantees when the metric and sequential entropies are of the same order of magnitude. In particular our algorithm is the first one that achieves optimal rates for online regression over Hölder balls. In addition we show for this example how to adapt our chaining algorithm to get a reasonable computational efficiency with similar regret guarantees (up to a log factor).

Keywords: online learning, nonparametric regression, chaining, individual sequences.

1. Introduction

We consider the setting of online nonparametric regression for arbitrary deterministic sequences, which unfolds as follows. First, the environment chooses a sequence of observations $(y_t)_{t\geqslant 1}$ in $\mathbb{R}$ and a sequence of input vectors $(x_t)_{t\geqslant 1}$ in $\mathcal{X}$, both initially hidden from the forecaster. At each time instant $t \in \mathbb{N}^* = \{1, 2, \dots\}$, the environment reveals the data $x_t \in \mathcal{X}$; the forecaster then gives a prediction $\widehat{y}_t \in \mathbb{R}$; the environment in turn reveals the observation $y_t \in \mathbb{R}$; and finally, the forecaster incurs the square loss $(y_t - \widehat{y}_t)^2$.

The term online *nonparametric* regression means that we are interested in forecasters whose regret

$$\mathrm{Reg}_T(\mathcal{F}) \triangleq \sum_{t=1}^{T} \bigl(y_t - \widehat{y}_t\bigr)^2 - \inf_{f\in\mathcal{F}} \sum_{t=1}^{T} \bigl(y_t - f(x_t)\bigr)^2$$

over standard nonparametric function classes $\mathcal{F} \subseteq \mathbb{R}^{\mathcal{X}}$ is as small as possible. In this paper we design and study an algorithm that achieves a regret bound of the form

$$\mathrm{Reg}_T(\mathcal{F}) \leqslant c_1 B^2 \bigl(1 + \log \mathcal{N}_\infty(\mathcal{F}, \gamma)\bigr) + c_2 B\sqrt{T} \int_0^{\gamma} \sqrt{\log \mathcal{N}_\infty(\mathcal{F}, \varepsilon)}\, d\varepsilon\ , \tag{1}$$

where $\gamma \in \left(\frac{B}{T}, B\right)$ is a parameter of the algorithm, where B is an upper bound on $\max_{1\leqslant t\leqslant T} |y_t|$, and where $\log \mathcal{N}_\infty(\mathcal{F}, \varepsilon)$ denotes the metric entropy of the function set $\mathcal{F}$ in the sup norm at scale ε (cf. Section 1.4).

The integral on the right-hand side of (1) is very close to what is known in probability theory as *Dudley's entropy integral*, a useful tool to upper bound the expectation of a centered stochastic process with subgaussian increments (see, e.g., Talagrand 2005; Boucheron et al. 2013). In statistical learning (with i.i.d. data), Dudley's entropy integral is key to derive risk bounds on empirical risk minimizers; see, e.g., Massart (2007); Rakhlin et al. (2013).

Very recently Rakhlin and Sridharan (2014) showed that the same type of entropy integral appears naturally in regret bounds for online nonparametric regression. The most part of their analysis is non-constructive in the sense that their regret bounds are obtained without explicitly constructing an algorithm. (Though they provide an abstract relaxation recipe for algorithmic purposes, we were not able to turn it into an explicit algorithm for online regression over nonparametric classes such as Hölder balls.)

One of our main contributions is to provide an explicit algorithm that achieves the regret bound (1). We note however that our regret bounds are in terms of a weaker notion of entropy, namely, metric entropy instead of the smaller (and optimal) sequential entropy. Fortunately, both notions are of the same order of magnitude for a reasonable number of examples, such as the ones outlined just below. We leave the question of modifying our algorithm to get sequential entropy regret bounds for future work.

The regret bound (1)—that we call *Dudley-type regret bound* thereafter—can be used to obtain optimal regret bounds for several classical nonparametric function classes. Indeed, when $\mathcal{F}$ has a metric entropy $\log \mathcal{N}_\infty(\mathcal{F}, \varepsilon) \leqslant C_p \varepsilon^{-p}$ with[1] $p \in (0,2)$, the bound (1) entails

$$\begin{aligned}\operatorname{Reg}_T(\mathcal{F}) &\leqslant c_1 B^2 + c_1 B^2 C_p \gamma^{-p} + c_2 B \sqrt{C_p T} \int_0^\gamma \varepsilon^{-p/2} d\varepsilon \\ &= c_1 B^2 + c_1 B^2 C_p \gamma^{-p} + \frac{2\, c_2 B}{2-p} \sqrt{C_p T}\, \gamma^{1-p/2} = \mathcal{O}\big(T^{p/(p+2)}\big) \qquad (2)\end{aligned}$$

for the choice of $\gamma = \Theta\big(T^{-1/(p+2)}\big)$. An example is given by Hölder classes $\mathcal{F}$ with regularity $\beta > 1/2$ (cf. Tsybakov 2009, Def 1.2). We know from (Kolmogorov and Tikhomirov, 1961) or (Lorentz, 1962, Theorem 2) that they satisfy $\log \mathcal{N}_\infty(\mathcal{F}, \varepsilon) = \mathcal{O}\big(\varepsilon^{-1/\beta}\big)$. Therefore, (2) entails a regret bound $\operatorname{Reg}_T(\mathcal{F}) = \mathcal{O}\big(T^{1/(2\beta+1)}\big)$, which is in a way optimal since it corresponds to the optimal (minimax) quadratic risk $T^{-2\beta/(2\beta+1)}$ in statistical estimation with i.i.d. data.

1.1. Why a simple Exponentially Weighted Average forecaster is not sufficient

A natural approach (see Vovk 2006) to compete against a nonparametric class $\mathcal{F}$ relies in running an Exponentially Weighted Average forecaster (EWA, see Cesa-Bianchi and Lugosi 2006, p.14) on an ε-net $\mathcal{F}^{(\varepsilon)}$ of $\mathcal{F}$ of finite size $\mathcal{N}_\infty(\mathcal{F}, \varepsilon)$. This yields a regret bound of order $\varepsilon T + \log \mathcal{N}_\infty(\mathcal{F}, \varepsilon)$. The first term εT is due to the approximation of $\mathcal{F}$ by $\mathcal{F}^{(\varepsilon)}$, while the second term is the regret suffered by EWA on the finite class of experts $\mathcal{F}^{(\varepsilon)}$. As noted by Rakhlin and Sridharan (2014, Remark 11), the above regret bound is suboptimal for large nonparametric classes $\mathcal{F}$. Indeed, for a metric entropy of order ε^{-p} with $p \in (0,2)$, optimizing the above regret bound in ε entails a regret of order $\mathcal{O}(T^{p/(p+1)})$ when (1) yields the better rate $\mathcal{O}(T^{p/(p+2)})$.

1.2. Constructing an online algorithm via the chaining technique

Next we explain how the chaining technique from Dudley (1967) (see appendix A for a brief reminder) can be used to build an algorithm that satisfies a Dudley-type regret bound (1). We approximate any function $f \in \mathcal{F}$ by a sequence of refining approximations $\pi_0(f) \in \mathcal{F}^{(0)}, \pi_1(f) \in \mathcal{F}^{(1)}, \ldots$, such that for all $k \geqslant 0$, $\sup_f \|\pi_k(f) - f\|_\infty \leqslant \gamma/2^k$ and $\operatorname{card} \mathcal{F}^{(k)} = \mathcal{N}_\infty(\mathcal{F}, \gamma/2^k)$, so

[1] When $p > 2$, we can also derive Dudley-type regret bounds that lead to a regret of $\mathcal{O}\big(T^{1-1/p}\big)$ in the same spirit as in Rakhlin and Sridharan (2014). We omitted this case to ease the presentation.

that:

$$\inf_{f\in\mathcal{F}}\sum_{t=1}^{T}\bigl(y_t - f(x_t)\bigr)^2 = \inf_{f\in\mathcal{F}}\sum_{t=1}^{T}\Biggl(y_t - \pi_0(f)(x_t) - \sum_{k=0}^{\infty}\underbrace{\bigl[\pi_{k+1}(f)-\pi_k(f)\bigr](x_t)}_{\|\cdot\|_\infty\leqslant 3\gamma/2^{k+1}}\Biggr)^2 .$$

We use the above decomposition in Algorithm 2 (Section 2.2) by performing two simultaneous aggregation tasks at two different scales:

- high-scale aggregation: we run an Exponentially Weighted Average forecaster to be competitive against every function $\pi_0(f)$ in the coarsest set $\mathcal{F}^{(0)}$;
- low-scale aggregation: we run in parallel many instances of (an extension of) the Exponentiated Gradient (EG) algorithm so as to be competitive against the increments $\pi_{k+1}(f)-\pi_k(f)$. The advantage of using EG is that even if the number $N^{(k)}$ of increments $\pi_{k+1}(f)-\pi_k(f)$ is large for small scales ε, the size of the gradients is very small, hence a manageable regret.

At the core of the algorithm lies the Multi-variable Exponentiated Gradient algorithm (Algorithm 1) that makes it possible to perform low-scale aggregation at all scales $\varepsilon < \gamma$ simultaneously.

1.3. Comparison to previous works and main contributions

Earlier uses of chaining and related techniques Several ideas that we use in this paper were already present in the literature. Opper and Haussler (1997) and Cesa-Bianchi and Lugosi (2001) derived Dudley-type regret bounds for the log loss using a two-scale aggregation and chaining arguments. At small scales, their algorithm is very specific to the log loss and it is unclear how to extend it to other exp-concave loss functions such as the square loss. Besides, they only use the chaining technique in their analysis by reducing the regret to an expected supremum, in the same spirit as Rakhlin et al. (2013) (square loss, batch setting) and Rakhlin and Sridharan (2014) (square loss, online learning with individual sequences). On the contrary, Cesa-Bianchi and Lugosi (1999) built an algorithm via chaining ideas (they use discretization sets $\mathcal{F}^{(k)}$ similar to those above). However, their algorithm is specific to linear loss functions (e.g., absolute loss with binary observations), so that no linearization step and no high-scale aggregation are required.

Other papers on online learning with nonparametric classes Related works also include the paper by Vovk (2006) where—for the problem under consideration here—suboptimal regret bounds are derived with the Exponentially Weighted Average forecaster. Another example of paper that addressed online learning over nonparametric function classes is the one by Hazan and Megiddo (2007). They also studied the regret with respect to the set of Lipschitz functions on $[0,1]^d$, but their loss functions are Lipschitz, hence their slower rates compared to ours.

Main contributions and outline of the paper Our contributions are threefold: we first design the Multi-variable Exponentiated Gradient algorithm (Section 2.1) which is crucial for the linearization step at all small scales simultaneously. We then present our main algorithm and derive a Dudley-type regret bound as in (1) (Section 2.2). This general algorithm is computationally intractable for nonparametric classes. In Section 3 we design an efficient algorithm in the case of Hölder classes. To the best of our knowledge, this is the first time the chaining technique has been used in a concrete fashion for individual sequences. Some proofs are postponed to the appendix.

1.4. Some useful definitions

Let $\mathcal{F} \subseteq \mathbb{R}^{\mathcal{X}}$ be a set of bounded functions endowed with the sup norm $\|f\|_\infty \triangleq \sup_{x \in \mathcal{X}} |f(x)|$. For all $\varepsilon > 0$, we call *proper ε-net* any subset $\mathcal{G} \subseteq \mathcal{F}$ such that $\forall f \in \mathcal{F},\ \exists g \in \mathcal{G} : \ \|f-g\|_\infty \leqslant \varepsilon$. (If $\mathcal{G} \not\subseteq \mathcal{F}$, we call it *non-proper*.) The cardinality of the smallest proper ε-net is denoted by $\mathcal{N}_\infty(\mathcal{F}, \varepsilon)$, and the logarithm $\log \mathcal{N}_\infty(\mathcal{F}, \varepsilon)$ is called the *metric entropy of $\mathcal{F}$ at scale ε*. When this quantity is finite for all $\varepsilon > 0$, we say that $\big(\mathcal{F}, \|\cdot\|_\infty\big)$ is *totally bounded*.

2. The Chaining Exponentially Weighted Average Forecaster

In this section we design an online algorithm—the *Chaining Exponentially Weighted Average forecaster*—that achieves the Dudley-type regret bound (1). In Section 2.1 below, we first define a subroutine that will prove crucial in our analysis, and whose applicability may extend beyond this paper.

2.1. Preliminary: the Multi-variable Exponentiated Gradient Algorithm

Let $\Delta_N \triangleq \left\{ \boldsymbol{u} \in \mathbb{R}_+^N : \sum_{i=1}^N u_i = 1 \right\} \subseteq \mathbb{R}^N$ denote the simplex in $\mathbb{R}^N$. In this subsection we define and study a new extension of the Exponentiated Gradient algorithm (Kivinen and Warmuth, 1997; Cesa-Bianchi, 1999). This extension is meant to minimize a sequence of multi-variable loss functions $\big(\boldsymbol{u}^{(1)}, \ldots, \boldsymbol{u}^{(K)}\big) \mapsto \ell_t\big(\boldsymbol{u}^{(1)}, \ldots, \boldsymbol{u}^{(K)}\big)$ simultaneously over all the variables $(\boldsymbol{u}^{(1)}, \ldots, \boldsymbol{u}^{(K)}) \in \Delta_{N_1} \times \ldots \times \Delta_{N_K}$.

Our algorithm is described as Algorithm 1 below. We call it *Multi-variable Exponentiated Gradient*. When $K = 1$, it boils down to the classical Exponentiated Gradient algorithm over the simplex Δ_{N_1}. But when $K \geqslant 2$, it performs K simultaneous optimization updates (one for each direction $\boldsymbol{u}^{(k)}$) that lead to a global optimum by joint convexity of the loss functions ℓ_t.

Algorithm 1: Multi-variable Exponentiated Gradient

```
input          : optimization domain $\Delta_{N_1} \times \ldots \times \Delta_{N_K}$ and tuning parameters $\eta^{(1)}, \ldots, \eta^{(K)} > 0$.
initialization: set $\widehat{\boldsymbol{u}}_1^{(k)} \triangleq \left(\frac{1}{N_k}, \ldots, \frac{1}{N_k}\right) \in \Delta_{N_k}$ for all $k = 1, \ldots, K$.
for each round $t = 1, 2, \ldots$ do
    • Output $\big(\widehat{\boldsymbol{u}}_t^{(1)}, \ldots, \widehat{\boldsymbol{u}}_t^{(K)}\big) \in \Delta_{N_1} \times \ldots \times \Delta_{N_K}$ and observe the differentiable and jointly
      convex loss function $\ell_t : \Delta_{N_1} \times \ldots \times \Delta_{N_K} \to \mathbb{R}$.
    • Compute the new weight vectors $\big(\widehat{\boldsymbol{u}}_{t+1}^{(1)}, \ldots, \widehat{\boldsymbol{u}}_{t+1}^{(K)}\big) \in \Delta_{N_1} \times \ldots \times \Delta_{N_K}$ as follows:
          $\widehat{u}_{t+1,i}^{(k)} \triangleq \dfrac{\exp\left(-\eta^{(k)} \sum_{s=1}^{t} \partial_{\widehat{u}_{s,i}^{(k)}} \ell_s\left(\widehat{\boldsymbol{u}}_s^{(1)}, \ldots, \widehat{\boldsymbol{u}}_s^{(K)}\right)\right)}{Z_{t+1}^{(k)}}, \quad i \in \{1, \ldots, N_k\},$
      where $\partial_{\widehat{u}_{s,i}^{(k)}} \ell_s$ is the partial derivative of $\ell_s$ with respect to the $i$-th component of $\widehat{\boldsymbol{u}}_s^{(k)}$, and
      where the normalizing factor is $Z_{t+1}^{(k)} \triangleq \sum_{i=1}^{N_k} \exp\left(-\eta^{(k)} \sum_{s=1}^{t} \partial_{\widehat{u}_{s,i}^{(k)}} \ell_s\left(\widehat{\boldsymbol{u}}_s^{(1)}, \ldots, \widehat{\boldsymbol{u}}_s^{(K)}\right)\right)$.
end
```

The Multi-variable Exponentiated Gradient algorithm satisfies the regret bound of Theorem 1 below. We first need some notations. We define the *partial gradients*

$$\nabla_{\boldsymbol{u}^{(k)}} \ell_t = \left(\partial_{u_1^{(k)}} \ell_t, \ldots, \partial_{u_{N_k}^{(k)}} \ell_t \right) , \qquad 1 \leqslant k \leqslant K ,$$

where $\partial_{u_i^{(k)}} \ell_t$ denotes the partial derivative of ℓ_t with respect to the scalar variable $u_i^{(k)}$. Note that $\nabla_{\boldsymbol{u}^{(k)}} \ell_t$ is a function that maps $\Delta_{N_1} \times \ldots \times \Delta_{N_K}$ to $\mathbb{R}^{N_k}$. Next we also use the notation

$$\|\varphi\|_\infty \triangleq \sup_{\boldsymbol{u}^{(1)},\ldots,\boldsymbol{u}^{(K)}} \max_{1 \leqslant i \leqslant N_k} \left| \varphi_i\left(\boldsymbol{u}^{(1)}, \ldots, \boldsymbol{u}^{(K)}\right) \right|$$

for the sup norm of any vector-valued function $\varphi : \Delta_{N_1} \times \ldots \times \Delta_{N_K} \to \mathbb{R}^{N_k}$, $1 \leqslant k \leqslant K$.

Theorem 1 *Assume that the loss functions $\ell_t : \Delta_{N_1} \times \ldots \times \Delta_{N_K} \to \mathbb{R}$, $t \geqslant 1$, are differentiable and jointly convex. Assume also the following upper bound on their partial gradients: for all $k \in \{1, \ldots, K\}$,*

$$\max_{1 \leqslant t \leqslant T} \|\nabla_{\boldsymbol{u}^{(k)}} \ell_t\|_\infty \leqslant G^{(k)} . \tag{3}$$

Then, the Multi-variable Exponentiated Gradient algorithm (Algorithm 1) tuned with the parameters $\eta^{(k)} = \sqrt{2 \log(N_k)/T} / G^{(k)}$ has a regret bounded as follows:

$$\sum_{t=1}^{T} \ell_t\left(\widehat{\boldsymbol{u}}_t^{(1)}, \ldots, \widehat{\boldsymbol{u}}_t^{(K)}\right) - \min_{\boldsymbol{u}^{(1)},\ldots,\boldsymbol{u}^{(K)}} \sum_{t=1}^{T} \ell_t\left(\boldsymbol{u}^{(1)}, \ldots, \boldsymbol{u}^{(K)}\right) \leqslant \sqrt{2T} \sum_{k=1}^{K} G^{(k)} \sqrt{\log N_k} ,$$

where the minimum is taken over all $\left(\boldsymbol{u}^{(1)}, \ldots, \boldsymbol{u}^{(K)}\right) \in \Delta_{N_1} \times \ldots \times \Delta_{N_K}$.

The proof of Theorem 1 is postponed to Appendix D.1.

2.2. The Chaining Exponentially Weighted Average Forecaster

In this section we introduce our main algorithm: the *Chaining Exponentially Weighted Average forecaster*. A precise definition will be given in Algorithm 2 below. For the sake of clarity, we first describe the main ideas underlying this algorithm.

Recall that we aim at proving a regret bound of the form (1), whose right-hand side consists of two main terms:

$$B^2 \log \mathcal{N}_\infty(\mathcal{F}, \gamma) \qquad \text{and} \qquad B\sqrt{T} \int_0^\gamma \sqrt{\log \mathcal{N}_\infty(\mathcal{F}, \varepsilon)} d\varepsilon .$$

Our algorithm performs aggregation at two different levels: one level (at all scales $\varepsilon \in (0, \gamma]$) to get the entropy integral above, and another level (at scale γ) to get the other term $B^2 \log \mathcal{N}_\infty(\mathcal{F}, \gamma)$. More precisely:

- for all $k \in \mathbb{N}$, let $\mathcal{F}^{(k)}$ be a proper $\gamma/2^k$-net of $(\mathcal{F}, \|\cdot\|_\infty)$ of minimal cardinality[2] $\mathcal{N}_\infty\left(\mathcal{F}, \gamma/2^k\right)$;
- for all $k \geqslant 1$, set $\mathcal{G}^{(k)} \triangleq \{\pi_k(f) - \pi_{k-1}(f) : \ f \in \mathcal{F}\}$, where

$$\forall f \in \mathcal{F}, \quad \pi_k(f) \in \operatorname{argmin}_{h \in \mathcal{F}^{(k)}} \|f - h\|_\infty .$$

We denote:

- the elements of $\mathcal{F}^{(0)}$ by $f_1^{(0)}, \ldots, f_{N_0}^{(0)}$ with $N_0 = \mathcal{N}_\infty\left(\mathcal{F}, \gamma\right)$;

[2] We assume that $(\mathcal{F}, \|\cdot\|_\infty)$ is totally bounded.

- the elements of $\mathcal{G}^{(k)}$ by $g_1^{(k)}, \ldots, g_{N_k}^{(k)}$; note that $N_k \leqslant \mathcal{N}_\infty\big(\mathcal{F}, \gamma/2^k\big)\mathcal{N}_\infty\big(\mathcal{F}, \gamma/2^{k-1}\big)$.

With the above definitions, our algorithm can be described as follows:

1. Low-scale aggregation: for every $j \in \{1, \ldots, N_0\}$, we use a Multi-variable Exponentiated Gradient forecaster to mimic the best predictor in the neighborhood of $f_j^{(0)}$: we set, at each round $t \geqslant 1$,
$$\widehat{f}_{t,j} \triangleq f_j^{(0)} + \sum_{k=1}^{K} \sum_{i=1}^{N_k} \widehat{u}_{t,i}^{(j,k)} g_i^{(k)} \,, \tag{4}$$
where $K \triangleq \lceil \log_2(\gamma T/B) \rceil$, so that the lowest scale is $\gamma/2^K \approx B/T$. The above weight vectors $\widehat{u}_t^{(j,k)} \in \Delta_{N_k}$ are defined in Equation (6) of Algorithm 2. They correspond exactly to the weight vectors output by the Multi-variable Exponentiated Gradient forecaster (Algorithm 1) applied to the loss functions $\ell_t^{(j)} : \Delta_{N_1} \times \ldots \times \Delta_{N_K} \to \mathbb{R}$ defined for all $t \geqslant 1$ (j is fixed) by
$$\ell_t^{(j)}\Big(\boldsymbol{u}^{(1)}, \ldots, \boldsymbol{u}^{(K)}\Big) = \left(y_t - f_j^{(0)}(x_t) - \sum_{k=1}^{K} \sum_{i=1}^{N_k} u_i^{(k)} g_i^{(k)}(x_t)\right)^2 \,. \tag{5}$$

2. High-scale aggregation: we use a standard Exponentially Weighted Average forecaster to aggregate all the $\widehat{f}_{t,j}$, $j = 1, \ldots, N_0$, as follows:
$$\widehat{f}_t = \sum_{j=1}^{N_0} \widehat{w}_{t,j} \widehat{f}_{t,j} \,,$$
where the weights $\widehat{w}_{t,j}$ are defined in Equation (7) of Algorithm 2. At time t, our algorithm predicts y_t with $\widehat{y}_t \triangleq \widehat{f}_t(x_t)$.

Next we show that the Chaining Exponentially Weighted Average forecaster satisfies a Dudley-type regret bound as in (1).

Theorem 2 *Let $B > 0$, $T \geqslant 1$, and $\gamma \in \big(B/T, B\big)$.*

- *Assume that $\max_{1 \leqslant t \leqslant T} |y_t| \leqslant B$ and that $\sup_{f \in \mathcal{F}} \|f\|_\infty \leqslant B$.*
- *Assume that $(\mathcal{F}, \|\cdot\|_\infty)$ is totally bounded and define $\mathcal{F}^{(0)} = \big\{f_1^{(0)}, \ldots, f_{N_0}^{(0)}\big\}$ and $\mathcal{G}^{(k)} = \big\{g_1^{(k)}, \ldots, g_{N_k}^{(k)}\big\}$, $k = 1, \ldots, K$, as above.*

Then, the Chaining Exponentially Weighted Average forecaster (Algorithm 2) tuned with the parameters $\eta^{(0)} = 1/(50B^2)$ and $\eta^{(k)} = \sqrt{2\log(N_k)/T}\, 2^k/(30B\gamma)$ for all $k = 1, \ldots, K$ satisfies:
$$\mathrm{Reg}_T(\mathcal{F}) \leqslant B^2\big(5 + 50 \log \mathcal{N}_\infty(\mathcal{F}, \gamma)\big) + 120B\sqrt{T} \int_0^{\gamma/2} \sqrt{\log \mathcal{N}_\infty(\mathcal{F}, \varepsilon)} d\varepsilon \,.$$

As a corollary (cf. (2) in the introduction), when $\log \mathcal{N}_\infty(\mathcal{F}, \varepsilon) \leqslant C_p \varepsilon^{-p}$ with $p \in (0, 2)$, the Chaining Exponentially Weighted Average forecaster tuned as above and with $\gamma = \Theta\big(T^{-1/(p+2)}\big)$ has a regret of $\mathcal{O}\big(T^{p/(p+2)}\big)$. This in turn yields a regret of $\mathrm{Reg}_T(\mathcal{F}) = \mathcal{O}\big(T^{1/(2\beta+1)}\big)$ when $\mathcal{F}$ is the Hölder class with regularity $\beta > 1/2$, which corresponds to the optimal (minimax) quadratic risk $T^{-2\beta/(2\beta+1)}$ in statistical estimation with i.i.d. data. We address the particular case of Hölder functions and the associated computational issues in Section 3 and Appendix C below.

Algorithm 2: Chaining Exponentially Weighted Average forecaster

```
input         : maximal range $B > 0$, tuning parameters $\eta^{(0)}, \eta^{(1)}, \ldots, \eta^{(K)} > 0$,
                high-scale functions $f_j^{(0)} : \mathcal{X} \to \mathbb{R}$ for $1 \leqslant j \leqslant N_0$,
                low-scale functions $g_i^{(k)} : \mathcal{X} \to \mathbb{R}$ for $k \in \{1, \ldots, K\}$ and $i \in \{1, \ldots, N_k\}$.
initialization: set $\widehat{\boldsymbol{w}}_1 = \big(\frac{1}{N_0}, \ldots, \frac{1}{N_0}\big) \in \Delta_{N_0}$ and
                $\widehat{\boldsymbol{u}}_1^{(j,k)} \triangleq \big(\frac{1}{N_k}, \ldots, \frac{1}{N_k}\big) \in \Delta_{N_k}$ for all $j \in \{1, \ldots, N_0\}$ and $k \in \{1, \ldots, K\}$.
for each round $t = 1, 2, \ldots$ do
    • Define the aggregated functions $\widehat{f}_{t,j} : \mathcal{X} \to \mathbb{R}$ for all $j \in \{1, \ldots, N_0\}$ by
          $$\widehat{f}_{t,j} \triangleq f_j^{(0)} + \sum_{k=1}^{K} \sum_{i=1}^{N_k} \widehat{u}_{t,i}^{(j,k)} g_i^{(k)} \,.$$
    • Observe $x_t \in \mathcal{X}$, predict $\widehat{y}_t = \sum_{j=1}^{N_0} \widehat{w}_{t,j} \widehat{f}_{t,j}(x_t)$, and observe $y_t \in [-B, B]$.
    • Low-scale update: compute the new weight vectors $\widehat{\boldsymbol{u}}_{t+1}^{(j,k)} = \Big(\widehat{u}_{t+1,i}^{(j,k)}\Big)_{1 \leqslant i \leqslant N_k} \in \Delta_{N_k}$ for all
      $j \in \{1, \ldots, N_0\}$ and $k \in \{1, \ldots, K\}$ as follows:
          $$\widehat{u}_{t+1,i}^{(j,k)} \triangleq \frac{\exp\left(-\eta^{(k)} \sum_{s=1}^{t} -2\Big(y_s - \widehat{f}_{s,j}(x_s)\Big) g_i^{(k)}(x_s)\right)}{\sum_{i'=1}^{N_k} \exp\left(-\eta^{(k)} \sum_{s=1}^{t} -2\Big(y_s - \widehat{f}_{s,j}(x_s)\Big) g_{i'}^{(k)}(x_s)\right)} \,, \quad i \in \{1, \ldots, N_k\} \,. \tag{6}$$
    • High-scale update: compute the new weight vector $\widehat{\boldsymbol{w}}_{t+1} = \big(\widehat{w}_{t+1,j}\big)_{1 \leqslant j \leqslant N_0} \in \Delta_{N_0}$ as
      follows:
          $$\widehat{w}_{t+1,j} \triangleq \frac{\exp\left(-\eta^{(0)} \sum_{s=1}^{t} \Big(y_s - \widehat{f}_{s,j}(x_s)\Big)^2\right)}{\sum_{j'=1}^{N_0} \exp\left(-\eta^{(0)} \sum_{s=1}^{t} \Big(y_s - \widehat{f}_{s,j'}(x_s)\Big)^2\right)} \,, \quad j \in \{1, \ldots, N_0\} \,. \tag{7}$$
end
```

Another corollary of Theorem 2 can be drawn in the setting of sparse high-dimensional online linear regression, which is a particular case of a parametric class with $p \approx 0$. In the same spirit as in Gerchinovitz (2013) and in Rakhlin and Sridharan (2014, Example 1), we consider d features $\varphi_1, \ldots, \varphi_d : \mathcal{X} \to [-B, B]$ and we define $\mathcal{F} = \big\{\sum_{j=1}^{d} u_j \varphi_j : \ \boldsymbol{u} \in \Delta_d, \|\boldsymbol{u}\|_0 = s\big\}$ to be the set of all s-sparse convex combinations of the features ($\|\boldsymbol{u}\|_0$ denotes the number of non-zero coefficients of $\boldsymbol{u}$). Then, using Theorem 1 of Gao et al. (2013) with $(M, q, p, r) = (s, 1, +\infty, 2)$ we can see that $\log \mathcal{N}_\infty(\mathcal{F}, \varepsilon) \lesssim \log \binom{d}{s} + s \log\big(1 + 1/(\varepsilon\sqrt{s})\big)$. Plugging this bound in Theorem 2 with $\gamma = 1/\sqrt{T}$ yields a regret bound of order $s \log(1 + dT/s)$. Thus, Theorem 2 also yields (quasi) optimal rates for sparse high-dimensional online linear regression.

Finally, for much larger function classes with $p > 2$, we could derive a slightly modified Dudley-type regret bound of $\mathcal{O}\big(T^{1-1/p}\big)$ with a slightly modified algorithm, in the same spirit as in Rakhlin and Sridharan (2014). We omit this case for the sake of conciseness.

Remark 3 *In Theorem 2 above, we assumed that the observations y_t and the predictions $f(x_t)$ are all bounded by B, and that B is known in advance by the forecaster. We can actually remove this requirement by using adaptive techniques of Gerchinovitz and Yu (2014), namely, adaptive clipping of the intermediate predictions $\widehat{f}_{t,j}(x_t)$ and adaptive Lipschitzification of the square loss functions $\ell_t^{(j)}$. This modification enables us to derive the same regret bound (up to multiplicative constant factors) with $B = \max_t |y_t|$, but without knowing B in advance, and without requiring that $\sup_{f\in\mathcal{F}} \|f\|_\infty$ is also upper bounded by B. Of course these adaptation techniques also make it possible to tune all parameters without knowing T in advance.*

Remark 4 *Even in the case when B is known by the forecaster, the clipping and Lipschitzification techniques of Gerchinovitz and Yu (2014) can be useful to get smaller constants in the regret bound. We could indeed replace the constants 50 and 120 with 8 and 48 respectively. (Moreover, the regret bound would also hold true for $\gamma > B$.) We chose however not to use these refinements in order to simplify the analysis.*

Remark 5 *We assumed that the performance of a forecast $\widehat{y}_t$ at round $t \geqslant 1$ is measured through the square loss $\ell_t(\widehat{y}_t) = (\widehat{y}_t - y_t)^2$, which is $1/(50B^2)$-exp-concave on $[-4B, 4B]$. The analysis can easily be extended to all η-exp-concave (and thus convex) loss functions ℓ_t on $[-4B, 4B]$ that also satisfy a self-bounding property of the form $\big|d\ell_t/d\widehat{y}_t\big| \leqslant C\ell_t^r$ (an example is given by $\ell_t(\widehat{y}_t) = \big|\widehat{y}_t - y_t\big|^r$ with $r \geqslant 2$). The regret bound of Theorem 2 remains unchanged up to a multiplicative factor depending on B, C, and r. If the loss functions ℓ_t are only convex (e.g., the absolute loss $\ell_t(\widehat{y}_t) = |\widehat{y}_t - y_t|$ or the pinball loss to perform quantile regression), the high-scale aggregation step is more costly: the term of order $\log \mathcal{N}_\infty(\mathcal{F}, \gamma)$ is replaced with a term of order $\sqrt{T \log \mathcal{N}_\infty(\mathcal{F}, \gamma)}$.*

Proof (of Theorem 2) We split our proof into two parts—one for each aggregation level.

Part 1: low-scale aggregation.

In this part, we fix $j \in \{1, \ldots, N_0\}$. As explained right before (5), the tuple of weight vectors $\big(\widehat{\boldsymbol{u}}_t^{(j,1)}, \ldots, \widehat{\boldsymbol{u}}_t^{(j,K)}\big) \in \Delta_{N_1} \times \ldots \Delta_{N_K}$ computed at all rounds $t \geqslant 1$ corresponds exactly to the output of the Multi-variable Exponentiated Gradient forecaster when applied to the loss functions $\ell_t^{(j)}$, $t \geqslant 1$, defined in (5). We can therefore apply Theorem 1 after checking its assumptions:

- the loss functions $\ell_t^{(j)}$ are indeed differentiable and jointly convex;
- the norms $\big\|\nabla_{\widehat{\boldsymbol{u}}_t^{(j,k)}} \ell_t^{(j)}\big\|_\infty$ of the partial gradients are bounded by $30B\gamma/2^k$ for all $1 \leqslant k \leqslant K$. Indeed, the i-th coordinate of $\nabla_{\widehat{\boldsymbol{u}}_t^{(j,k)}} \ell_t^{(j)}$ is equal to
$$\partial_{\widehat{u}_{t,i}^{(j,k)}} \ell_t^{(j)} = -2\Big(y_t - \widehat{f}_{t,j}(x_t)\Big) g_i^{(k)}(x_t), \tag{8}$$
which can be upper bounded (in absolute value) by $2 \times 5B \times 3\gamma/2^k$. To see why this is true, first note that $\big|g_i^{(k)}(x_t)\big| \leqslant \big\|g_i^{(k)}\big\|_\infty = \big\|\pi_k(f) - \pi_{k-1}(f)\big\|_\infty$ for some $f \in \mathcal{F}$ (by definition of $\mathcal{G}^{(k)}$), so that, by the triangle inequality and by definition of $\pi_k(f)$ and $\mathcal{F}^{(k)}$:
$$\Big|g_i^{(k)}(x_t)\Big| \leqslant \big\|\pi_k(f) - f\big\|_\infty + \big\|\pi_{k-1}(f) - f\big\|_\infty \leqslant \frac{\gamma}{2^k} + \frac{\gamma}{2^{k-1}} = \frac{3\gamma}{2^k}\,. \tag{9}$$

Second, note that $\bigl|y_t - \widehat{f}_{t,j}(x_t)\bigr| \leqslant |y_t| + \bigl|\widehat{f}_{t,j}(x_t)\bigr| \leqslant 5B$. Indeed, we have $|y_t| \leqslant B$ by assumption and, by definition of $\widehat{f}_{t,j}$ in (4), we also have

$$\bigl|\widehat{f}_{t,j}(x_t)\bigr| \leqslant \Bigl\| f_j^{(0)} \Bigr\|_\infty + \sum_{k=1}^{K} \sum_{i=1}^{N_k} \widehat{u}_{t,i}^{(j,k)} \bigl|g_i^{(k)}(x_t)\bigr| \leqslant B + \sum_{k=1}^{K} \frac{3\gamma}{2^k} \leqslant B + 3\gamma \leqslant 4B\,, \qquad (10)$$

where we used the inequalities $\bigl\|f_j^{(0)}\bigr\|_\infty \leqslant \sup_{f \in \mathcal{F}} \|f\|_\infty \leqslant B$ (by assumption), and where we combined (9) with the fact that $\sum_{i=1}^{N_k} \widehat{u}_{t,i}^{(j,k)} = 1$. The last inequality above is obtained from the assumption $\gamma \leqslant B$. Substituting the above various upper bounds in (8) entails that $\bigl\|\nabla_{\widehat{\boldsymbol{u}}_t^{(j,k)}} \ell_t^{(j)}\bigr\|_\infty \leqslant 30B\gamma/2^k$ for all $1 \leqslant k \leqslant K$, as claimed earlier.

We are now in a position to apply Theorem 1. It yields:

$$\sum_{t=1}^{T} \Bigl(y_t - \widehat{f}_{t,j}(x_t)\Bigr)^2 \leqslant \inf_{g_1,\dots,g_K} \sum_{t=1}^{T} \Bigl(y_t - \Bigl(f_j^{(0)} + g_1 + \ldots + g_K\Bigr)(x_t)\Bigr)^2$$
$$+ \sqrt{2T} \sum_{k=1}^{K} 30B\gamma/2^k \sqrt{\log N_k}\,, \qquad (11)$$

where the infimum is over all functions $g_1 \in \mathcal{G}^{(1)}, \dots, g_K \in \mathcal{G}^{(K)}$ (we used the regret bound of Theorem 1 with Dirac weight vectors $\boldsymbol{u}^{(k)} = \delta_{i_k}$, $i_k = 1, \dots, N_k$).

Now, using the fact that $N_k \leqslant \mathcal{N}_\infty(\mathcal{F}, \gamma/2^k)\mathcal{N}_\infty(\mathcal{F}, \gamma/2^{k-1}) \leqslant \bigl(\mathcal{N}_\infty(\mathcal{F}, \gamma/2^k)\bigr)^2$, we get

$$\sum_{k=1}^{K} \frac{\gamma}{2^k} \sqrt{\log N_k} \leqslant 2\sqrt{2} \sum_{k=1}^{K} \left(\frac{\gamma}{2^k} - \frac{\gamma}{2^{k+1}}\right) \sqrt{\log \mathcal{N}_\infty(\mathcal{F}, \gamma/2^k)}$$
$$\leqslant 2\sqrt{2} \sum_{k=1}^{K} \int_{\gamma/2^{k+1}}^{\gamma/2^k} \sqrt{\log \mathcal{N}_\infty(\mathcal{F}, \varepsilon)}\, d\varepsilon \leqslant 2\sqrt{2} \int_0^{\gamma/2} \sqrt{\log \mathcal{N}_\infty(\mathcal{F}, \varepsilon)}\, d\varepsilon\,,$$

where the inequality before last follows by monotonicity of $\varepsilon \mapsto \mathcal{N}_\infty(\mathcal{F}, \varepsilon)$ on every interval $\bigl[\gamma/2^{k+1}, \gamma/2^k\bigr]$. Finally, substituting the above integral in (11) yields

$$\sum_{t=1}^{T} \Bigl(y_t - \widehat{f}_{t,j}(x_t)\Bigr)^2 \leqslant \inf_{g_1,\dots,g_K} \sum_{t=1}^{T} \Bigl(y_t - \Bigl(f_j^{(0)} + g_1 + \ldots + g_K\Bigr)(x_t)\Bigr)^2$$
$$+ 120B\sqrt{T} \int_0^{\gamma/2} \sqrt{\log \mathcal{N}_\infty(\mathcal{F}, \varepsilon)}\, d\varepsilon\,. \qquad (12)$$

Part 2: high-scale aggregation.

The prediction $\widehat{y}_t = \widehat{f}_t(x_t) = \sum_{j=1}^{N_0} \widehat{w}_{t,j} \widehat{f}_{t,j}(x_t)$ at time t is a convex combination of the intermediate predictions $\widehat{f}_{t,j}(x_t)$, where the weights $\widehat{w}_{t,j}$ correspond exactly to those of the standard Exponentially Weighted Average forecaster tuned with $\eta^{(0)} = 1/(50B^2) = 1/\bigl(2(5B)^2\bigr)$. Since the intermediate predictions $\widehat{f}_{t,j}(x_t)$ lie in $[-4B, 4B]$ (by (10) above), and since the square loss $z \mapsto (y_t - z)^2$ is $\eta^{(0)}$-exp-concave on $[-4B, 4B]$ for any $y_t \in [-B, B]$, we get from Proposition 3.1 and Page 46 of Cesa-Bianchi and Lugosi (2006) that

$$\sum_{t=1}^{T} (y_t - \widehat{y}_t)^2 \leqslant \min_{1 \leqslant j \leqslant N_0} \sum_{t=1}^{T} \Big(y_t - \widehat{f}_{t,j}(x_t) \Big)^2 + \frac{\log N_0}{\eta^{(0)}}$$

$$\leqslant \inf_{f_0, g_1, \ldots, g_K} \sum_{t=1}^{T} \big(y_t - (f_0 + g_1 + \ldots + g_K)(x_t) \big)^2$$

$$+ 120 B\sqrt{T} \int_0^{\gamma/2} \sqrt{\log \mathcal{N}_\infty(\mathcal{F}, \varepsilon)}\, d\varepsilon + 50 B^2 \log \mathcal{N}_\infty(\mathcal{F}, \gamma) \ , \qquad (13)$$

where the infimum is over all functions $f_0 \in \mathcal{F}^{(0)}, g_1 \in \mathcal{G}^{(1)}, \ldots, g_K \in \mathcal{G}^{(K)}$. The last inequality above was a consequence of (12). Next we apply the chaining idea: by definition of the function sets $\mathcal{F}^{(0)} \supseteq \{\pi_0(f) : \ f \in \mathcal{F}\}$ and $\mathcal{G}^{(k)} = \{\pi_k(f) - \pi_{k-1}(f) : \ f \in \mathcal{F}\}$, we have

$$\inf_{f_0, g_1, \ldots, g_K} \sum_{t=1}^{T} \big(y_t - (f_0 + g_1 + \ldots + g_K)(x_t) \big)^2$$

$$\leqslant \inf_{f \in \mathcal{F}} \sum_{t=1}^{T} \Big(y_t - \big(\pi_0(f) + [\pi_1(f) - \pi_0(f)] + \ldots + [\pi_K(f) - \pi_{K-1}(f)] \big)(x_t) \Big)^2$$

$$= \inf_{f \in \mathcal{F}} \sum_{t=1}^{T} \big(y_t - \pi_K(f)(x_t) \big)^2$$

$$\leqslant \inf_{f \in \mathcal{F}} \sum_{t=1}^{T} \Big[\big(y_t - f(x_t) \big)^2 + 2 \cdot 2B \left\| \pi_K(f) - f \right\|_\infty + \left\| \pi_K(f) - f \right\|_\infty^2 \Big] \qquad (14)$$

$$\leqslant \inf_{f \in \mathcal{F}} \sum_{t=1}^{T} \big(y_t - f(x_t) \big)^2 + 4B^2 + \frac{B^2}{T} \ , \qquad (15)$$

where (14) is obtained by expanding the square $\big(y_t - \pi_K(f)(x_t)\big)^2 = \big(y_t - f(x_t) + f(x_t) - \pi_K(f)(x_t)\big)^2$, and where (15) follows from the fact that $\left\|\pi_K(f) - f\right\|_\infty \leqslant \gamma/2^K \leqslant B/T$ by definition of $\pi_K(f)$ and $K = \lceil \log_2(\gamma T/B) \rceil$. Combining (13) and (15) concludes the proof. ■

3. An efficient chaining algorithm for Hölder classes

The Chaining Exponentially Weighted Average forecaster of the previous section is quite natural since it explicitly exploits the ε-nets that appear in the Dudley-type regret bound (1). However its time and space computational complexities are prohibitively large (exponential in T) since it is necessary to update exponentially many weights at every round t. It actually turns out that, fortunately, most standard function classes have a sufficiently nice structure. This enables us to adapt the previous chaining technique on (quasi-optimal) ε-nets that are much easier to exploit from an algorithmic viewpoint. We describe below the particular case of Lipschitz classes; the more general case of Hölder classes is postponed to the appendix.

In all the sequel, $\mathcal{F}$ denotes the set of functions from $[0,1]$ to $[-B, B]$ that are 1-Lipschitz. Recall from the introduction that $\log \mathcal{N}_\infty(\mathcal{F}, \varepsilon) = \Theta(\varepsilon^{-1})$, so that, by Theorem 2 and (2), the Chaining Exponentially Weighted Average forecaster guarantees a regret of $\mathcal{O}\big(T^{1/3}\big)$. We explain below how to modify this algorithm with ε-nets of $\big(\mathcal{F}, \| \cdot \|_\infty\big)$ that are easier to manage from a computational viewpoint. This leads to a quasi-optimal regret of $\mathcal{O}\big(T^{1/3} \log T\big)$; see Theorem 6.

3.1. Constructing computationally-manageable ε-nets via a dyadic discretization

Let $\gamma \in \left(\frac{B}{T}, B\right)$ be a fixed real number that will play the same role as in Theorem 2. Using the fact that all functions in $\mathcal{F}$ are 1-Lipschitz, we can approximate $\mathcal{F}$ with piecewise-constant functions as follows. We partition the x-axis $[0,1]$ into $1/\gamma$ subintervals $I_a \triangleq \big[(a-1)\gamma, a\gamma\big)$, $a = 1, \ldots, 1/\gamma$ (the last interval is closed at $x = 1$). We also use a discretization of length γ on the y-axis $[-B, B]$, by considering values of the form $c^{(0)} = -B + j\gamma$, $j = 0, \ldots, 2B/\gamma$. (For the sake of simplicity, we assume that both $1/\gamma$ and $2B/\gamma$ are integers.) We then define the set $\mathcal{F}^{(0)}$ of piecewise-constant functions $f^{(0)} : [0,1] \to [-B, B]$ of the form

$$f^{(0)}(x) = \sum_{a=1}^{1/\gamma} c_a^{(0)} \mathbb{I}_{x \in I_a} \,, \qquad c_1^{(0)}, \ldots, c_{1/\gamma}^{(0)} \in \mathcal{C}^{(0)} \triangleq \left\{ -B + j\gamma : j = 0, \ldots, \frac{2B}{\gamma} \right\} . \tag{16}$$

Using the fact that all functions in $\mathcal{F}$ are 1-Lipschitz, it is quite straightforward to see that $\mathcal{F}^{(0)}$ is a γ-net[3] of $\big(\mathcal{F}, \|\cdot\|_\infty\big)$. (To see why this is true, we can choose $c_a^{(0)} \in \operatorname{argmin}_{c \in C^{(0)}} \big|f(x_a) - c\big|$, where x_a is the center of the subinterval I_a. See Lemma 13 in the appendix for further details.)

Refinement via a dyadic discretization Next we construct $\gamma/2^m$-nets that are refinements of the γ-net $\mathcal{F}^{(0)}$. We need to define a dyadic discretization for each subinterval I_a as follows: for any level $m \geqslant 1$, we partition I_a into 2^m subintervals $I_a^{(m,n)}$, $n = 1, \ldots, 2^m$, of equal size $\gamma/2^m$. Note that the subintervals $I_a^{(m,n)}$, $a = 1, \ldots, 1/\gamma$ and $n = 1, \ldots, 2^m$, form a partition of $[0,1]$. We call it *the level-m partition*. We enrich the set $\mathcal{F}^{(0)}$ by looking at all the functions of the form $f^{(0)} + \sum_{m=1}^M g^{(m)}$, where $f^{(0)} \in \mathcal{F}^{(0)}$ and where every function $g^{(m)}$ is piecewise-constant on the level-m partition, with values $c_a^{(m,n)} \in \big[-\gamma/2^{m-1}, \gamma/2^{m-1}\big]$ that are small when m is large. In other words, we define *the level-M approximation set* $\mathcal{F}^{(M)}$ as the set of all functions $f_{\boldsymbol{c}} : [0,1] \to \mathbb{R}$ of the form

$$f_{\boldsymbol{c}}(x) = \underbrace{\sum_{a=1}^{1/\gamma} c_a^{(0)} \mathbb{I}_{x \in I_a}}_{f^{(0)}(x)} + \sum_{m=1}^{M} \underbrace{\sum_{a=1}^{1/\gamma} \sum_{n=1}^{2^m} c_a^{(m,n)} \mathbb{I}_{x \in I_a^{(m,n)}}}_{g^{(m)}(x)} \,, \tag{17}$$

where $c_a^{(0)} \in \mathcal{C}^{(0)}$ and $c_a^{(m,n)} \in \big[-\gamma/2^{m-1}, \gamma/2^{m-1}\big]$. An example of function $f_{\boldsymbol{c}} = f^{(0)} + \sum_{m=1}^M g^{(m)}$ is plotted on Figure 1 in the case when $M = 2$ (the plot is restricted to the interval I_a).

Since all functions in $\mathcal{F}$ are 1-Lipschitz, the set $\mathcal{F}^{(M)}$ of all functions $f_{\boldsymbol{c}}$ is a $\gamma/2^{M+1}$-net of $(\mathcal{F}, \|\cdot\|_\infty)$; see Lemma 13 in the appendix for a proof. Note that $\mathcal{F}^{(M)}$ is infinite (the $c_a^{(m,n)}$ are continuously valued); fortunately this is not a problem since the $c_a^{(m,n)}$ can be rewritten as convex combinations $c_a^{(m,n)} = u_1^{(m,n)}(-\gamma/2^{m-1}) + u_2^{(m,n)}(\gamma/2^{m-1})$ of only two values; cf. (18) below.

3.2. A chaining algorithm using this dyadic discretization

Next we design an algorithm which, as in Section 2.2, is able to be competitive against any function $f_{\boldsymbol{c}} = f^{(0)} + \sum_{m=1}^M g^{(m)}$. However, instead of maintaining exponentially many weights as in Algorithm 2, we use the dyadic discretization in a crucial way. More precisely:

We run $1/\gamma$ instances of the same algorithm $\mathcal{A}$ in parallel; the a-th instance $\mathcal{A}_a$, $a = 1, \ldots, 1/\gamma$, corresponds to the subinterval I_a and it is updated only at rounds t such that $x_t \in I_a$.

[3] This γ-net is not proper since $\mathcal{F}^{(0)} \not\subseteq \mathcal{F}$.

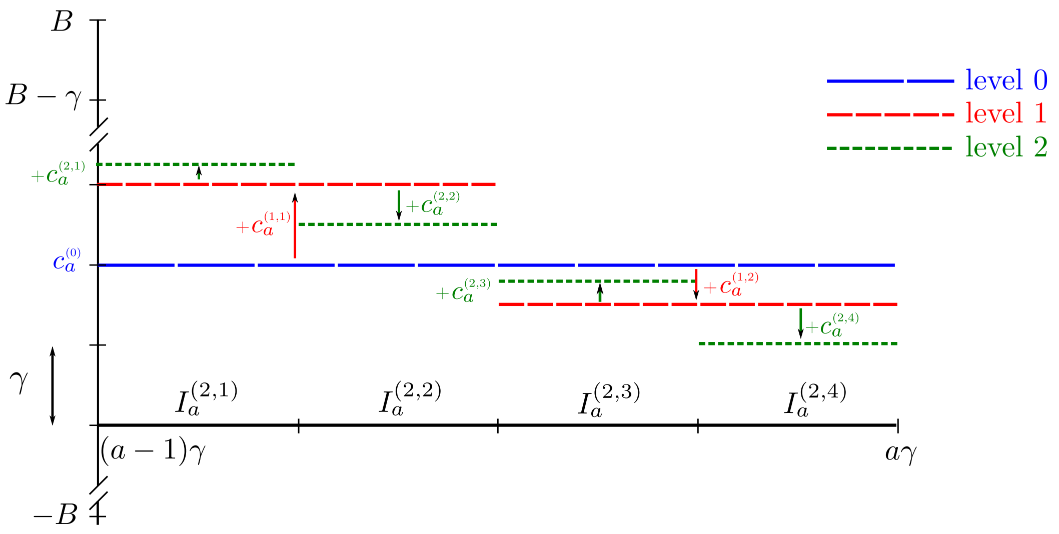

Figure 1: An example of function $f^{(0)} + \sum_{m=1}^{M} g^{(m)}$ for $M = 2$, plotted on the subinterval I_a. This function corresponds to the dotted line (level 2).

Next we focus on subalgorithm $\mathcal{A}_a$. As in Algorithm 2, we use a combination of the EWA and the Multi-variable EG forecasters to perform high-scale and low-scale aggregation simultaneously:

Low-scale aggregation: we run $2B/\gamma + 1$ instances $\mathcal{B}_{a,j}$, $j = 0, \dots, 2B/\gamma$, of the Adaptive Multi-variable Exponentiated Gradient algorithm (Algorithm 3 in the appendix) simultaneously. Each instance $\mathcal{B}_{a,j}$ corresponds to a particular constant $c^{(0)} = -B + j\gamma \in \mathcal{C}^{(0)}$ and is run (similarly to (5)) with the loss function ℓ_t defined for all weight vectors $\boldsymbol{u}^{(m,n)} = \left(u_1^{(m,n)}, u_2^{(m,n)}\right) \in \Delta_2$ by

$$\ell_t\left(\boldsymbol{u}^{(m,n)},\, m = 1, \dots, M,\, n = 1, \dots, 2^m\right) = \left(y_t - (-B + j\gamma) - \sum_{m=1}^{M}\sum_{n=1}^{2^m}\left(u_1^{(m,n)}\frac{-\gamma}{2^{m-1}} + u_2^{(m,n)}\frac{\gamma}{2^{m-1}}\right)\mathbb{I}_{x_t \in I_a^{(m,n)}}\right)^2 . \qquad (18)$$

The above convex combinations $u_1^{(m,n)}(-\gamma/2^{m-1}) + u_2^{(m,n)}(\gamma/2^{m-1})$ ensure that subalgorithm $\mathcal{B}_{a,j}$ is competitive against the best constants $c_a^{(m,n)} \in \left[-\gamma/2^{m-1}, \gamma/2^{m-1}\right]$ for all m and n.

The weight vectors output by subalgorithm $\mathcal{B}_{a,j}$ (when $x_t \in I_a$) are denoted by $\widehat{\boldsymbol{u}}_{t,a,j}^{(m,n)}$, and we set $\widehat{f}_{t,a,j}(x) \triangleq -B + j\gamma + \sum_{m=1}^{M}\sum_{n=1}^{2^m}\left(\widehat{u}_{t,a,j,1}^{(m,n)}\frac{-\gamma}{2^{m-1}} + \widehat{u}_{t,a,j,2}^{(m,n)}\frac{\gamma}{2^{m-1}}\right)\mathbb{I}_{x \in I_a^{(m,n)}}$ for all $j = 0, \dots, 2B/\gamma$.

High-scale aggregation: we aggregate the $2B/\gamma + 1$ forecasters above with a standard Exponentially Weighted Average forecaster (tuned, e.g., with the parameter $\eta = 1/(2(4B)^2) = 1/(32B^2)$):

$$\widehat{f}_{t,a} = \sum_{j=0}^{2B/\gamma} \widehat{w}_{t,a,j}\widehat{f}_{t,a,j} \, . \qquad (19)$$

Putting all things together: at every time $t \geqslant 1$, we make the prediction $\widehat{f}_t(x_t) \triangleq \sum_{a=1}^{1/\gamma} \widehat{f}_{t,a}(x_t)\mathbb{I}_{x_t \in I_a}$. We call this algorithm the *Dyadic Chaining Algorithm*.

Theorem 6 *Let $B > 0$, $T \geqslant 2$, and $\mathcal{F}$ be the set of all 1-Lipschitz functions from $[0,1]$ to $[-B,B]$. Assume that $\max_{1 \leqslant t \leqslant T} |y_t| \leqslant B$. Then, the Dyadic Chaining Algorithm defined above and tuned* *with the parameters* $\gamma = BT^{-1/3}$ *and* $M = \lceil \log_2(\gamma T/B) \rceil$ *satisfies, for some absolute constant* $c > 0$,

$$\mathrm{Reg}_T(\mathcal{F}) \leqslant c \max\{B, B^2\} T^{1/3} \log T \; .$$

The proof is postponed to the appendix. Note that the Dyadic Chaining Algorithm is computationally tractable: at every round t, the point x_t only falls into one subinterval $I_a^{(m,n)}$ for each level $m = 1, \ldots, M$, so that we only need to update $\mathcal{O}\big(2B/\gamma \times M\big) = \mathcal{O}\big(T^{1/3} \log T\big)$ weights at every round. For the same reason, the overall space complexity is $\mathcal{O}(T \times 2B/\gamma \times M) = \mathcal{O}\big(T^{4/3} \log T\big)$.

Remark 7 *The algorithm can be extended to the case of Lipschitz functions on* $[0,1]^d$*. It leads to an optimal regret of order* $\mathcal{O}(T^{d/(2+d)})$ *up to a log factor. Besides, the computational complexity is still tractable. Indeed, at each round t, the point* x_t *only falls into one cell of the partition. Hence, the time complexity is polynomial in* T *with an exponent independent of d. This also applies to the space complexity if we use sparsity-tailored data types. The extension to Hölder function classes on* $[0,1]^d$ *is however more difficult and we leave it for future work.*

Acknowledgments

The authors would like to thank Alexander S. Rakhlin for insightful initial discussions that triggered this work. The authors acknowledge the support of the French Agence Nationale de la Recherche (ANR), under grants ANR-13-BS01-0005 (project SPADRO) and ANR-13-CORD-0020 (project ALICIA).

References

S. Boucheron, G. Lugosi, and P. Massart. *Concentration inequalities: a nonasymptotic theory of independence*. Oxford University Press, 2013.

N. Cesa-Bianchi. Analysis of two gradient-based algorithms for on-line regression. *J. Comput. System Sci.*, 59(3):392–411, 1999.

N. Cesa-Bianchi and G. Lugosi. On prediction of individual sequences. *Ann. Statist.*, 27:1865–1895, 1999.

N. Cesa-Bianchi and G. Lugosi. Worst-case bounds for the logarithmic loss of predictors. *Mach. Learn.*, 43:247–264, 2001.

N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

R.M. Dudley. The sizes of compact subsets of hilbert space and continuity of gaussian processes. *Journal of Functional Analysis*, 1(3):290 – 330, 1967.

F. Gao, C.-K. Ing, and Y. Yang. Metric entropy and sparse linear approximation of ℓ_q-hulls for $0 < q \leqslant 1$. *J. Approx. Theory*, 166:42–55, 2013.

S. Gerchinovitz. *Prediction of individual sequences and prediction in the statistical framework: some links around sparse regression and aggregation techniques*. PhD thesis, Université Paris-Sud 11, Orsay, 2011.

S. Gerchinovitz. Sparsity regret bounds for individual sequences in online linear regression. *J. Mach. Learn. Res.*, 14:729–769, 2013.

S. Gerchinovitz and J.Y. Yu. Adaptive and optimal online linear regression on ℓ^1-balls. *Theoretical Computer Science*, 519:4–28, 2014.

E. Hazan and N. Megiddo. Online learning with prior knowledge. In N. H. Bshouty and C. Gentile, editors, *Proceedings of the 20th Annual Conference on Learning Theory (COLT'07)*, volume 4539 of *Lecture Notes in Computer Science*, pages 499–513. Springer Berlin Heidelberg, 2007.

J. Kivinen and M.K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Inform. and Comput.*, 132(1):1–63, 1997.

A.N. Kolmogorov and V.M. Tikhomirov. ε-entropy and ε-capacity of sets in function spaces. *Translations of the American Mathematical Society*, 17:277–364, 1961.

G.G. Lorentz. Metric entropy, widths, and superpositions of functions. *Amer. Math. Monthly*, 69 (6):469–485, 1962.

P. Massart. *Concentration Inequalities and Model Selection*, volume 1896 of *Lecture Notes in Mathematics*. Springer, Berlin, 2007.

M. Opper and D. Haussler. Worst case prediction over sequences under log loss. In *The Mathematics of Information Coding, Extraction, and Distribution*. Spinger Verlag, 1997.

A. Rakhlin and K. Sridharan. Online nonparametric regression. *JMLR W&CP*, 35 (Proceedings of COLT 2014):1232–1264, 2014.

A. Rakhlin, K. Sridharan, and A.B. Tsybakov. Empirical entropy, minimax regret and minimax risk. *Bernoulli*, 2013. URL http://arxiv.org/abs/1308.1147. To appear.

M. Talagrand. *The generic chaining*. Springer, 2005.

A.B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer, 2009.

V. Vovk. Metric entropy in competitive on-line prediction. *arXiv*, 2006. URL http://arxiv.org/abs/cs.LG/0609045.

Appendix A. The chaining technique: a brief reminder

The idea of chaining was introduced by Dudley (1967). It provides a general method to bound the supremum of stochastic processes. For the convenience of the reader, we recall the main ideas underlying this technique; see, e.g., Boucheron et al. (2013) for further details. We consider a centered stochastic process $(X_f)_{f\in\mathcal{F}}$ indexed by some finite metric space, say, $(\mathcal{F}, \|\cdot\|_\infty)$, with subgaussian increments, which means that $\log \mathbb{E} e^{\lambda(X_f - X_g)} \leqslant \frac{1}{2} v\lambda^2 \|f-g\|_\infty^2$ for all $\lambda > 0$ and all $f, g \in \mathcal{F}$. The goal is to bound the quantity $\mathbb{E}\bigl[\sup_{f\in\mathcal{F}} X_f\bigr] = \mathbb{E}\bigl[\sup_{f\in\mathcal{F}} (X_f - X_{f_0})\bigr]$ for any $f_0 \in \mathcal{F}$.

Lemma 8 (Boucheron et al. 2013) *Let $Z_1, \ldots, Z_K$ be subgaussian random variables with parameter $v > 0$ (i.e., $\log \mathbb{E} \exp(\lambda Z_i) \leqslant \lambda^2 v/2$ for all $\lambda \in \mathbb{R}$), then $\mathbb{E} \max_{i=1,\ldots,K} Z_i \leqslant \sqrt{2v \log K}$.*

Lemma 8 entails $\mathbb{E}\bigl[\sup_{f\in\mathcal{F}}(X_f - X_{f_0})\bigr] \leqslant B\sqrt{2v\log(\operatorname{card}\mathcal{F})}$, where $B = \sup_{f\in\mathcal{F}} \|f - f_0\|_\infty$. However, this bound is too crude since X_f and X_g are very correlated when f and g are very close. The chaining technique takes this remark into account by approximating the maximal value $\sup_f X_f$ by maxima over successive refining discretizations $\mathcal{F}^{(0)}, \ldots, \mathcal{F}^{(K)}$ of $\mathcal{F}$. More formally, for any $f \in \mathcal{F}$, we consider a sequence of approximations $\pi_0(f) = f_0 \in \mathcal{F}^{(0)}, \pi_1(f) \in \mathcal{F}^{(1)}, \ldots, \pi_K(f) = f \in \mathcal{F}^{(K)}$, where $\|f - \pi_k(f)\|_\infty \leqslant B/2^k$ and $\operatorname{card}\mathcal{F}^{(k)} = \mathcal{N}_\infty(\mathcal{F}, B/2^k)$, so that:

$$\mathbb{E}\Bigl[\sup_{f\in\mathcal{F}}(X_f - X_{f_0})\Bigr] = \mathbb{E}\left[\sup_{f\in\mathcal{F}} \sum_{k=0}^{K-1} \Bigl(X_{\pi_{k+1}(f)} - X_{\pi_k(f)}\Bigr)\right] \leqslant \sum_{k=0}^{K-1} \mathbb{E}\left[\sup_{f\in\mathcal{F}} \Bigl(X_{\pi_{k+1}(f)} - X_{\pi_k(f)}\Bigr)\right],$$

We apply Lemma 8 for each $k \in \{0, \ldots, K-1\}$: since $\|\pi_{k+1}(f) - \pi_k(f)\|_\infty \leqslant 3B/2^{k+1}$ (by the triangle inequality) and $\operatorname{card}\{\pi_{k+1}(f) - \pi_k(f), f \in \mathcal{F}\} \leqslant \mathcal{N}_\infty(\mathcal{F}, B/2^{k+1})^2$, we get the well-known Dudley entropy bound (note that $\varepsilon \mapsto \mathcal{N}_\infty(\mathcal{F}, \varepsilon)$ is nonincreasing):

$$\mathbb{E}\Bigl[\sup_{f\in\mathcal{F}}(X_f - X_{f_0})\Bigr] \leqslant 6\sum_{k=0}^{K-1} B2^{-k-1}\sqrt{v\log\mathcal{N}_\infty(\mathcal{F}, B/2^{k+1})} \leqslant 12\sqrt{v}\int_0^{B/2} \sqrt{\log\mathcal{N}_\infty(\mathcal{F},\varepsilon)}d\varepsilon\,.$$

Appendix B. Adaptive Multi-variable Exponentiated Gradient

In this subsection, we provide an adaptive version of Algorithm 1 when the time horizon T is not known in advance. We adopt the notations of Section 2.1. Basically, the fixed tuning parameters $\eta^{(1)}, \dots, \eta^{(k)}$ are replaced with time-varying learning rates $\eta_t^{(1)}, \dots, \eta_t^{(k)}$.

Algorithm 3: Adaptive Multi-variable Exponentiated Gradient

```
input         : optimization domain $\Delta_{N_1} \times \ldots \times \Delta_{N_K}$ (where $N_1, \dots, N_K$ are positive integers).
initialization: set $\widehat{\boldsymbol{u}}_1^{(k)} \triangleq \left(\frac{1}{N_k}, \dots, \frac{1}{N_k}\right) \in \Delta_{N_k}$ for all $k = 1, \dots, K$.
for each round $t = 1, 2, \dots$ do
    • Output $\left(\widehat{\boldsymbol{u}}_t^{(1)}, \dots, \widehat{\boldsymbol{u}}_t^{(K)}\right) \in \Delta_{N_1} \times \ldots \times \Delta_{N_K}$ and observe the differentiable and jointly
      convex loss function $\ell_t : \Delta_{N_1} \times \ldots \times \Delta_{N_K} \to \mathbb{R}$.
    • Update the tuning parameters, $\eta_t^{(k)}$ for all $k = 1, \dots, K$ as follows:
          $\eta_{t+1}^{(k)} = \frac{1}{G^{(k)}} \sqrt{\frac{\log N^{(k)}}{1 + \sum_{s=1}^{t} \mathbb{I}_{\left\|\nabla_{\boldsymbol{u}^{(k)}} \ell_s\right\|_\infty > 0}}}$
    • Compute the new weight vectors $\left(\widehat{\boldsymbol{u}}_{t+1}^{(1)}, \dots, \widehat{\boldsymbol{u}}_{t+1}^{(K)}\right) \in \Delta_{N_1} \times \ldots \times \Delta_{N_K}$ as follows:
          $\widehat{\boldsymbol{u}}_{t+1,i}^{(k)} \triangleq \frac{\exp\left(-\eta_{t+1}^{(k)} \sum_{s=1}^{t} \partial_{\widehat{u}_{s,i}^{(k)}} \ell_s\left(\widehat{\boldsymbol{u}}_s^{(1)}, \dots, \widehat{\boldsymbol{u}}_s^{(K)}\right)\right)}{Z_{t+1}^{(k)}}, \quad i \in \{1, \dots, N_k\},$
      where $\partial_{\widehat{u}_{s,i}^{(k)}} \ell_s$ denotes the partial derivative of $\ell_s$ with respect to $i$-th component of the vector
      variable $\widehat{\boldsymbol{u}}_s^{(k)}$, and where the normalization factor $Z_{t+1}^{(k)}$ is defined by
          $Z_{t+1}^{(k)} \triangleq \sum_{i=1}^{N_k} \exp\left(-\eta_{t+1}^{(k)} \sum_{s=1}^{t} \partial_{\widehat{u}_{s,i}^{(k)}} \ell_s\left(\widehat{\boldsymbol{u}}_s^{(1)}, \dots, \widehat{\boldsymbol{u}}_s^{(K)}\right)\right) .$
end
```

The Adaptive Multi-variable Exponentiated Gradient algorithm satisfies the regret bound of Theorem 9 below.

Theorem 9 *Assume that the loss functions $\ell_t : \Delta_{N_1} \times \ldots \times \Delta_{N_K} \to \mathbb{R}$, $t \geqslant 1$, are differentiable and jointly convex. Assume also the following upper bound on their partial gradients: for all $k \in \{1, \dots, K\}$,*

$$\max_{1 \leqslant t \leqslant T} \left\|\nabla_{\boldsymbol{u}^{(k)}} \ell_t\right\|_\infty \leqslant G^{(k)} . \tag{20}$$

Then, the Multi-variable Exponentiated Gradient algorithm (Algorithm 3) has a regret bounded as follows:

$$\sum_{t=1}^{T} \ell_t\Big(\widehat{\boldsymbol{u}}_t^{(1)}, \dots, \widehat{\boldsymbol{u}}_t^{(K)}\Big) - \min_{\boldsymbol{u}^{(1)},\dots,\boldsymbol{u}^{(K)}} \sum_{t=1}^{T} \ell_t\Big(\boldsymbol{u}^{(1)}, \dots, \boldsymbol{u}^{(K)}\Big) \leqslant 2\sum_{k=1}^{K} G^{(k)}\sqrt{T^{(k)}\log N_k}\,,$$

where $T^{(k)} = \sum_{t=1}^{T} \mathbb{I}_{\|\nabla_{\boldsymbol{u}^{(k)}}\ell_t\|_\infty > 0}$ *and where the minimum is taken over all* $\big(\boldsymbol{u}^{(1)}, \dots, \boldsymbol{u}^{(K)}\big) \in \Delta_{N_1} \times \dots \times \Delta_{N_K}$.

Proof (of Theorem 9) The proof starts as the one of Theorem 1. From (31), we can see that

$$\begin{aligned}
&\sum_{t=1}^{T} \ell_t\Big(\widehat{\boldsymbol{u}}_t^{(1)}, \dots, \widehat{\boldsymbol{u}}_t^{(K)}\Big) - \min_{\boldsymbol{u}^{(1)},\dots,\boldsymbol{u}^{(K)}} \sum_{t=1}^{T} \ell_t\Big(\boldsymbol{u}^{(1)}, \dots, \boldsymbol{u}^{(K)}\Big) \\
&\qquad = \sum_{k=1}^{K}\left(\sum_{t=1}^{T} \boldsymbol{g}_t^{(k)} \cdot \widehat{\boldsymbol{u}}_t^{(k)} - \min_{1\leqslant i\leqslant N_k} \sum_{t=1}^{T} g_{t,i}^{(k)}\right) \\
&\qquad = \sum_{k=1}^{K}\left(\sum_{t\in\mathcal{T}^{(k)}} \boldsymbol{g}_t^{(k)} \cdot \widehat{\boldsymbol{u}}_t^{(k)} - \min_{1\leqslant i\leqslant N_k} \sum_{t\in\mathcal{T}^{(k)}} g_{t,i}^{(k)}\right), \qquad (21)
\end{aligned}$$

where $\boldsymbol{g}_t^{(k)} \triangleq \nabla_{\widehat{\boldsymbol{u}}_t^{(k)}} \ell_t(\widehat{\boldsymbol{u}}_t^{(1)}, \dots, \widehat{\boldsymbol{u}}_t^{(K)})$ and where $\mathcal{T}^{(k)} = \big\{t = 1, \dots, T, \quad \|\nabla_{\boldsymbol{u}^{(k)}}\ell_t\|_\infty > 0\big\}$.

Note that the right-hand side of (31) is the sum of K regrets. Let $k \in \{1, \dots, K\}$. By definition of the Adaptive Multi-variable Exponentiated Gradient algorithm, the sequence of weight vectors $\big(\widehat{\boldsymbol{u}}_t^{(k)}\big)_{t\geqslant 1}$ corresponds exactly to the weight vectors output by the Exponentially Weighted Average forecaster with time-varying parameter (see Page 50 of Gerchinovitz 2011) applied to N_k experts associated with the loss vectors $\boldsymbol{g}_t^{(k)} \in \mathbb{R}^{N_k}$, $t \in \mathcal{T}^{(k)}$. We can therefore use the well-known corresponding regret bound available, e.g., in Proposition 2.1 of Gerchinovitz (2011). Noting that the loss vectors $\boldsymbol{g}_t^{(k)}$ lie in $\big[-G^{(k)}, G^{(k)}\big]^{N_k}$ by Assumption (20), and setting $T^{(k)} = \operatorname{card}\mathcal{T}^{(k)}$, we thus get that

$$\sum_{t\in\mathcal{T}^{(k)}} \boldsymbol{g}_t^{(k)} \cdot \widehat{\boldsymbol{u}}_t^{(k)} - \min_{1\leqslant i\leqslant N_k} \sum_{t\in\mathcal{T}^{(k)}} g_{t,i}^{(k)} \leqslant 2G^{(k)}\sqrt{T^{(k)}\log N_k}\,.$$

Note that the additional term $G^{(k)}\sqrt{\log N_k}$ in the upper-bound of Gerchinovitz (2011) is actually not needed, since we can assume that $\eta_{T+1}^{(k)} = \eta_T^{(k)}$ because $\eta_{T+1}^{(k)}$ is not used by the algorithm at rounds $t \leqslant T$. Substituting the last upper bound in the right-hand side of (21) concludes the proof. ■

Appendix C. An efficient chaining algorithm for Hölder classes

In this appendix, we extend the analysis of Section 3 to Hölder function classes. In the sequel $\mathcal{F}$ denotes the set of functions on $[0,1]$ whose q first derivatives ($q \in \mathbb{N}$) exist and are all bounded in supremum norm by a constant B, and whose qth derivative is Hölder continuous of order $\alpha \in (0,1]$ with coefficient $\lambda > 0$. In other words, any function $f \in \mathcal{F}$ satisfies

$$\forall x, y \in [0,1], \quad \bigl|f^{(q)}(x) - f^{(q)}(y)\bigr| \leqslant \lambda |x-y|^{\alpha}\,, \tag{22}$$

and $\|f^{(k)}\|_\infty \leqslant B$ for all $k \in \{0, \dots, q\}$. We denote by $\beta = q + \alpha$ the coefficient of regularity of $\mathcal{F}$. Recall from the introduction that $\log \mathcal{N}_\infty(\mathcal{F}, \varepsilon) = \mathcal{O}(\varepsilon^{-1/\beta})$, so that, by Theorem 2 and (2), if $\beta > 1/2$, the Chaining Exponentially Weighted Average forecaster guarantees a regret of $\mathcal{O}\bigl(T^{1/(2\beta+1)}\bigr)$, which is optimal. We explain below how to modify this algorithm with non-proper ε-nets of $(\mathcal{F}, \|\cdot\|_\infty)$ that are easier to manage from a computational viewpoint. This leads to a quasi-optimal regret of $\mathcal{O}\bigl(T^{1/(2\beta+1)}(\log T)^{3/2}\bigr)$.

The analysis follows the one of Section 3 which dealt with the special case of 1-Lipschitz functions. The main difference consists in replacing piecewise-constant approximations with piecewise-polynomial approximations.

C.1. Constructing computationally-manageable ε-nets via exponentially nested discretization

Let $\gamma \in \bigl(\frac{B}{T}, B\bigr)$ be a fixed real number that will play the same role as in Theorem 2. Using the fact that all functions in $\mathcal{F}$ are Hölder, we can approximate $\mathcal{F}$ with piecewise-polynomial functions as follows.

Let $\delta_x > 0$ and $\delta_y > 0$ be two discretization widths that will be fixed later by the analysis. We partition the x-axis $[0,1]$ into $1/\delta_x$ subintervals $I_a \triangleq \bigl[(a-1)\delta_x, a\delta_x\bigr)$, $a = 1, \dots, 1/\delta_x$ (the last interval is closed at $x = 1$). We also use a discretization of length δ_y on the y-axis $[-B, B]$, by considering the set

$$\mathcal{Y}^{(0)} \triangleq \Bigl\{ -B + j\delta_y \;\;:\;\; j = 0, \dots, 2B/\delta_y \Bigr\}\,.$$

For the sake of simplicity, we assume that both $1/\delta_x$ and $2B/\delta_y$ are integers. Otherwise, it suffices to consider $\lceil 1/\delta_x \rceil$ and $\lceil 2B/\delta_y \rceil$, which only impacts the constants of the final Theorem 11. We then define the sets of clipped polynomial functions for every $a \in \{1, \dots, 1/\delta_x\}$

$$\mathcal{P}_a^{(0)} \triangleq \left\{ x \mapsto \left[a_0 + \frac{a_1}{1!}(x - x_a)^2 + \cdots + \frac{a_q}{q!}(x - x_a)^q \right]_B \;\;:\;\; a_0, \dots, a_q \in \mathcal{Y}^{(0)} \right\}\,.$$

Here, $[\cdot]_B$ is the clipping operator defined by $[x]_B \triangleq \min\bigl\{B, \max\{-B, x\}\bigr\}$ and x_a is the center of I_a. Now, we define the set $\mathcal{F}^{(0)}$ of piecewise-clipped polynomial functions $f^{(0)} : [0,1] \to [-B, B]$ of the form

$$f^{(0)}(x) = \sum_{a=1}^{1/\delta_x} P_a^{(0)}(x) \mathbb{I}_{x \in I_a}\,, \qquad \forall a \in \{1, \dots, 1/\delta_x\} \quad P_a^{(0)} \in \mathcal{P}_a^{(0)}\,. \tag{23}$$

Remark that the above definition is similar to (16), where the constants $c_a^{(0)}$ have been substituted with clipped polynomials. Using the fact that all functions in $\mathcal{F}$ are Hölder, we can see (cf. Lemma 10) that for $\delta_x = 2\bigl(q!\gamma/(2\lambda)\bigr)^{1/\beta}$ and $\delta_y = \gamma/e$, the set $\mathcal{F}^{(0)}$ is a γ-net[4] of $\bigl(\mathcal{F}, \|\cdot\|_\infty\bigr)$.

[4] This γ-net is not proper since $\mathcal{F}^{(0)} \not\subseteq \mathcal{F}$.

Refinement via an exponentially nested discretization Next we construct $\gamma/2^m$-nets that are refinements of the γ-net $\mathcal{F}^{(0)}$. We need to define an exponentially nested discretization for each subinterval I_a as follows: for any level $m \geqslant 1$, we partition I_a into 4^m subintervals $I_a^{(m,n)}$, $n = 1, \dots, 4^m$, of equal size $\delta_x/4^m$. Note that the subintervals $I_a^{(m,n)}$, $a = 1, \dots, 1/\delta_x$ and $n = 1, \dots, 4^m$, form a partition of $[0,1]$. We call it *the level-m partition*.

Now, we design the sets of clipped polynomial functions $\mathcal{Q}_a^{(m,n)}$ that will refine the approximation of $\mathcal{F}$ on each interval $I_a^{(m,n)}$. To do so, for every $m \geqslant 1$ we set successive dyadic refining discretizations of the coefficients space $[-B, B]$:

$$\mathcal{Y}^{(m)} \triangleq \Big\{ -B + j\delta_y/2^m \ : \quad j = 0, \dots, 2^{m+1}B/\delta_y \Big\}, \tag{24}$$

and we define the corresponding sets of clipped polynomial functions for all $a \in \{1, \dots, 1/\delta_x\}$, all $m \in \{1, \dots, M\}$, and $n \in \{1, \dots, 4^m\}$

$$\mathcal{P}_a^{(m,n)} \triangleq \left\{ x \mapsto \left[a_0 + \frac{a_1}{1!}\left(x - x_a^{(m,n)}\right)^2 + \cdots + \frac{a_q}{q!}\left(x - x_a^{(m,n)}\right)^q \right]_B \ : \quad a_0, \dots, a_q \in \mathcal{Y}^{(m)} \right\}, \tag{25}$$

where $x_a^{(m,n)}$ is the center of the interval $I_a^{(m,n)}$. Then, we define the sets of differences between clipped polynomial functions of two consecutive levels

$$\mathcal{Q}_a^{(m,n)} = \left\{ \left[P^{(m)} - P^{(m-1)} \right]_{3\gamma/2^m} : P^{(m)} \in \mathcal{P}_a^{(m,n)} \text{ and } P^{(m-1)} \in \mathcal{P}_a^{(m-1,n_{m-1})} \right\}$$

where n_{m-1} denotes the unique integer n' such that $I_a^{(m,n)} \subset I_a^{(m-1,n')}$. (For $m = 1$, $\mathcal{P}_a^{(m-1,n_{m-1})}$ is replaced with $\mathcal{P}_a^{(0)}$ in the definition of $\mathcal{Q}_a^{(m,n)}$). The functions in $\mathcal{Q}_a^{(m,n)}$ will play the same role as the constants $c_a^{(m,n)}$ for the Lipschitz case to refine the approximation from the level-$(m-1)$ partition to the level-m partition. Note that each $Q_a^{(m,n)} \in \mathcal{Q}_a^{(m,n)}$ takes values in $[-3\gamma/2^m, 3\gamma/2^m]$.

Then, we enrich the set $\mathcal{F}^{(0)}$ by looking at all the functions of the form $f^{(0)} + \sum_{m=1}^M g^{(m)}$, where $f^{(0)} \in \mathcal{F}^{(0)}$ and where every function $g^{(m)}$ is the difference of a piecewise-clipped polynomial on the level-m partition and a piecewise-clipped polynomial on the previous level $m-1$, with values $Q_a^{(m,n)} \in \mathcal{Q}_a^{(m,n)}$.

In other words, we define *the level-M approximation set* $\mathcal{F}^{(M)}$ as the set of all functions $f_{\boldsymbol{c}} : [0,1] \to \mathbb{R}$ of the form

$$f_{\boldsymbol{c}}(x) = \underbrace{\sum_{a=1}^{1/\delta_x} P_a^{(0)}(x) \mathbb{I}_{x \in I_a}}_{f^{(0)}(x)} + \sum_{m=1}^{M} \underbrace{\sum_{a=1}^{1/\delta_x} \sum_{n=1}^{4^m} Q_a^{(m,n)}(x) \mathbb{I}_{x \in I_a^{(m,n)}}}_{g^{(m)}(x)}, \tag{26}$$

where $P_a^{(0)} \in \mathcal{P}_a^{(0)}$ and $Q_a^{(m,n)} \in \mathcal{Q}_a^{(m,n)}$. Once again, see (26) as an extension of (17), where the constants $c_a^{(m,n)}$ have been replaced with $Q_a^{(m,n)}$.

Using again the fact that all functions in $\mathcal{F}$ are Hölder, we can show that the set $\mathcal{F}^{(M)}$ of all functions $f_{\boldsymbol{c}}$ is a $\gamma/2^M$-net of $(\mathcal{F}, \|\cdot\|_\infty)$; see Lemma 10 below (whose proof is postponed to Appendix D.3) for further details.

Lemma 10 *Let $\mathcal{F}$ be the set of Hölder functions defined in (22). Assume that $\beta \triangleq q + \alpha \geqslant 1/2$. Let $\delta_x = 2(q!\gamma/(2\lambda))^{1/\beta}$ and $\delta_y = \gamma/e$. Then:*

- *the set $\mathcal{F}^{(0)}$ defined in (23) is a γ-net of $\big(\mathcal{F}, \|\cdot\|_\infty\big)$;*
- *for all $M \geqslant 1$, the set $\mathcal{F}^{(M)}$ defined in (26) is a $\gamma/2^M$-net of $\big(\mathcal{F}, \|\cdot\|_\infty\big)$.*

C.2. A chaining algorithm using this exponentially nested refining discretization

Next we design an algorithm which, as in Section 3, is able to be competitive against any function $f_{\boldsymbol{c}} = f^{(0)} + \sum_{m=1}^M g^{(m)}$ and is computationally tractable. More precisely:

We run $1/\delta_x$ instances of the same algorithm $\mathcal{A}$ in parallel; the a-th instance corresponds to the subinterval I_a and it is updated only at rounds t such that $x_t \in I_a$.

Next we focus on the a-th instance of the algorithm $\mathcal{A}$, whose local time is only incremented when a new x_t falls into I_a. As in Algorithm 2, we use a combination of the EWA and the Multi-variable EG forecasters to perform high-scale and low-scale aggregation simultaneously:

Low-scale aggregation: we run $\operatorname{card}\mathcal{P}_a^{(0)} \leqslant (2B/\delta_y+1)^{(q+1)}$ instances $\mathcal{B}_{a,j}$, $j = 1,\dots,\operatorname{card}\mathcal{P}_a^{(0)}$ of the Adaptive Multi-variable Exponentiated Gradient algorithm (Algorithm 3 in the appendix) simultaneously. Each instance $\mathcal{B}_{a,j}$ corresponds to a particular polynomial $P_{a,j}^{(0)} \in \mathcal{P}_a^{(0)}$ and is run (similarly to (5)) with the loss function ℓ_t defined for all weight vectors $\boldsymbol{u}^{(m,n)} \in \Delta_{\operatorname{card}\mathcal{Q}_a^{(m,n)}}$ by

$$\ell_t\left(\boldsymbol{u}^{(m,n)},\, m = 1,\dots,M,\, n = 1,\dots,4^m\right) = \left(y_t - P_{a,j}^{(0)}(x_t) - \sum_{m=1}^{M}\sum_{n=1}^{4^m}\sum_{k=1}^{\operatorname{card}\mathcal{Q}_a^{(m,n)}} u_k^{(m,n)} Q_{a,k}^{(m,n)}(x_t)\mathbb{I}_{x_t\in I_a^{(m,n)}}\right)^2 . \qquad (27)$$

Here, $Q_{a,1}^{(m,n)}, Q_{a,2}^{(m,n)}, \dots$ denote the elements of $\mathcal{Q}_a^{(m,n)}$ that have been ordered. The above convex combinations $\sum_k u_k^{(m,n)} Q_{a,k}^{(m,n)}$ ensure that subalgorithm $\mathcal{B}_{a,j}$ is competitive against the best elements in $\mathcal{Q}_a^{(m,n)}$ on subintervals $I_a^{(m,n)}$ for all m and n. The weight vectors formed by this subalgorithm $\mathcal{B}_{a,j}$ (when $x_t \in I_a$) are denoted by $\widehat{\boldsymbol{u}}_{t,a,j}^{(m,n)}$, and we set for all $j = 1,\dots,\operatorname{card}\mathcal{P}_a^{(0)}$

$$\widehat{f}_{t,a,j}(x) \triangleq P_{a,j}^{(0)}(x) + \sum_{m=1}^{M}\sum_{n=1}^{4^m}\sum_{k=1}^{\operatorname{card}\mathcal{Q}_a^{(m,n)}} \widehat{u}_{t,a,j,k}^{(m,n)}\, Q_{a,k}^{(m,n)}(x)\mathbb{I}_{x\in I_a^{(m,n)}} ,$$

where $P_{a,j}^{(0)}$ is the jth element of $\mathcal{P}_a^{(0)}$.

High-scale aggregation: we aggregate the forecasters above $\widehat{f}_{t,a,j}$ for $j \in \left\{1,\dots,\operatorname{card}\mathcal{P}_a^{(0)}\right\}$ with a standard Exponentially Weighted Average forecaster (tuned, e.g., with the parameter $\eta = 1/(2(5B)^2) = 1/(50B^2)$):

$$\widehat{f}_{t,a} = \sum_{j=1}^{\operatorname{card}\mathcal{P}_a^{(0)}} \widehat{w}_{t,a,j}\widehat{f}_{t,a,j} . \qquad (28)$$

Putting all things together: at every time $t \geqslant 1$, we make the prediction $\widehat{f}_t(x_t) \triangleq \sum_{a=1}^{1/\delta_x} \widehat{f}_{t,a}(x_t)\mathbb{I}_{x_t \in I_a}$. We call this algorithm the *Nested Chaining Algorithm for Hölder functions*.

Theorem 11 *Let $B > 0$, $T \geqslant 2$, and $\mathcal{F}$ be the set of Hölder functions defined in* (22). *Assume that $\beta \triangleq q + \alpha \geqslant 1/2$ and that $\max_{1 \leqslant t \leqslant T} |y_t| \leqslant B$. Then, the Nested Chaining Algorithm for Hölder functions defined above and tuned with the parameters $\delta_x = 2(q!\gamma/(2\lambda))^{1/\beta}$, $\delta_y = \gamma/e$, $\gamma = BT^{-\beta/(2\beta+1)}$ and $M = \lceil \log_2(\gamma T/B) \rceil$ satisfies, for some constant $c > 0$ depending only on q and λ,*

$$\mathrm{Reg}_T(\mathcal{F}) \leqslant c \max\{B^{2-1/\beta}, B^2\} T^{\frac{1}{2\beta+1}} (\log T)^{3/2} \,.$$

The proof is postponed to Appendix D.5. The logarithmic factor $(\log T)^{3/2}$ can be reduced to $\log T$, by partitioning I_a into $2^{m/\beta}$ subintervals $I_a^{(m,n)}$ instead of 4^m subintervals. However, the partition at level $m \geqslant 2$ is then not necessarily nested in the partitions of lower levels, which makes the proof slightly more difficult.

Note that the Nested Chaining Algorithm for Hölder functions is computationally tractable as shown by the following lemma, whose proof is deferred to Appendix D.6.

Lemma 12 *Under the assumptions of Theorem 11, the complexity of the Nested Chaining Algorithm for Hölder functions defined above satisfies:*

- *Storage complexity:* $\mathcal{O}\Big(T^{2q+4+\frac{\beta(q-1)+1}{2\beta+1}} \log T\Big)$ *;*

- *Time complexity:* $\mathcal{O}\left(T^{(q+1)\left(2+\frac{\beta}{2\beta+1}\right)} \log T\right)$.

Appendix D. Omitted proofs

In this appendix, we provide the proofs which were omitted in the main body of the paper.

D.1. Proof of Theorem 1

As is the case for the classical Exponentiated Gradient algorithm, the proof relies on a linearization argument. Let $\bigl(\boldsymbol{u}^{(1)},\dots,\boldsymbol{u}^{(K)}\bigr) \in \Delta_{N_1}\times\ldots\times\Delta_{N_K}$. By differentiability and joint convexity of ℓ_t for all $t=1,\dots,T$, we have that

$$\begin{aligned}
&\sum_{t=1}^{T}\ell_t\Bigl(\widehat{\boldsymbol{u}}_t^{(1)},\dots,\widehat{\boldsymbol{u}}_t^{(K)}\Bigr)-\sum_{t=1}^{T}\ell_t\Bigl(\boldsymbol{u}^{(1)},\dots,\boldsymbol{u}^{(K)}\Bigr)\\
&\qquad\leqslant\sum_{t=1}^{T}\nabla\ell_t\Bigl(\widehat{\boldsymbol{u}}_t^{(1)},\dots,\widehat{\boldsymbol{u}}_t^{(K)}\Bigr)\cdot\Bigl(\widehat{\boldsymbol{u}}_t^{(1)}-\boldsymbol{u}^{(1)},\dots,\widehat{\boldsymbol{u}}_t^{(K)}-\boldsymbol{u}^{(K)}\Bigr) &(29)\\
&\qquad=\sum_{t=1}^{T}\sum_{k=1}^{K}\nabla_{\widehat{\boldsymbol{u}}_t^{(k)}}\ell_t\Bigl(\widehat{\boldsymbol{u}}_t^{(1)},\dots,\widehat{\boldsymbol{u}}_t^{(K)}\Bigr)\cdot\Bigl(\widehat{\boldsymbol{u}}_t^{(k)}-\boldsymbol{u}^{(k)}\Bigr)\;, &(30)
\end{aligned}$$

where $\nabla\ell_t$ in (29) denotes the usual (joint) gradient of ℓ_t (with $\sum_{k=1}^{K}N_k$ components), and where (30) follows from splitting the gradient into K partial gradients: $\nabla\ell_t=\Bigl(\nabla_{\widehat{\boldsymbol{u}}_t^{(1)}}\ell_t,\dots,\nabla_{\widehat{\boldsymbol{u}}_t^{(K)}}\ell_t\Bigr)$.

As a consequence, setting $\boldsymbol{g}_t^{(k)}\triangleq\nabla_{\widehat{\boldsymbol{u}}_t^{(k)}}\ell_t\Bigl(\widehat{\boldsymbol{u}}_t^{(1)},\dots,\widehat{\boldsymbol{u}}_t^{(K)}\Bigr)\in\mathbb{R}^{N_k}$, and taking the maximum of the last inequality over all $\bigl(\boldsymbol{u}^{(1)},\dots,\boldsymbol{u}^{(K)}\bigr)\in\Delta_{N_1}\times\ldots\times\Delta_{N_K}$, we can see that

$$\begin{aligned}
&\sum_{t=1}^{T}\ell_t\Bigl(\widehat{\boldsymbol{u}}_t^{(1)},\dots,\widehat{\boldsymbol{u}}_t^{(K)}\Bigr)-\min_{\boldsymbol{u}^{(1)},\dots,\boldsymbol{u}^{(K)}}\sum_{t=1}^{T}\ell_t\Bigl(\boldsymbol{u}^{(1)},\dots,\boldsymbol{u}^{(K)}\Bigr)\\
&\qquad\leqslant\sum_{k=1}^{K}\max_{\boldsymbol{u}^{(k)}\in\Delta_{N_k}}\sum_{t=1}^{T}\boldsymbol{g}_t^{(k)}\cdot\Bigl(\widehat{\boldsymbol{u}}_t^{(k)}-\boldsymbol{u}^{(k)}\Bigr)\\
&\qquad=\sum_{k=1}^{K}\left(\sum_{t=1}^{T}\boldsymbol{g}_t^{(k)}\cdot\widehat{\boldsymbol{u}}_t^{(k)}-\min_{1\leqslant i\leqslant N_k}\sum_{t=1}^{T}g_{t,i}^{(k)}\right)\;, &(31)
\end{aligned}$$

where the last inequality follows from the fact that the function $\boldsymbol{u}^{(k)}\mapsto\sum_{t=1}^{T}\boldsymbol{g}_t^{(k)}\cdot\boldsymbol{u}^{(k)}$ is linear over the polytope Δ_{N_k}, so that its minimum is achieved on at least one of the N_k vertices of Δ_{N_k}.

Note that the right-hand side of (31) is the sum of K regrets. Let $k\in\{1,\dots,K\}$. By definition of the Multi-variable Exponentiated Gradient algorithm, the sequence of weight vectors $\bigl(\widehat{\boldsymbol{u}}_t^{(k)}\bigr)_{t\geqslant1}$ corresponds exactly to the weight vectors output by the Exponentially Weighted Average forecaster (see Page 14 of Cesa-Bianchi and Lugosi 2006) applied to N_k experts associated with the loss vectors $\boldsymbol{g}_t^{(k)}\in\mathbb{R}^{N_k}$, $t\geqslant1$. We can therefore use the well-known corresponding regret bound available, e.g., in Theorem 2.2 of Cesa-Bianchi and Lugosi (2006) or in Theorem 2.1 of Gerchinovitz (2011). Noting that the loss vectors $\boldsymbol{g}_t^{(k)}$ lie in $\bigl[-G^{(k)},G^{(k)}\bigr]^{N_k}$ by Assumption (3), we thus get that

$$\sum_{t=1}^{T}\boldsymbol{g}_t^{(k)}\cdot\widehat{\boldsymbol{u}}_t^{(k)}-\min_{1\leqslant i\leqslant N_k}\sum_{t=1}^{T}g_{t,i}^{(k)}\leqslant G^{(k)}\sqrt{2T\log N_k}\;.$$

substituting the last upper bound in the right-hand side of (31) concludes the proof.

D.2. An efficient γ-net for Lipschitz classes

Lemma 13 *Let $\mathcal{F}$ be the set of functions from $[0,1]$ to $[-B,B]$ that are 1-Lipschitz. Then:*

- *the set $\mathcal{F}^{(0)}$ defined in (16) is a γ-net of $\bigl(\mathcal{F}, \|\cdot\|_\infty\bigr)$;*
- *for all $M \geqslant 1$, the set $\mathcal{F}^{(M)}$ defined in (17) is a $\gamma/2^{M+1}$-net of $\bigl(\mathcal{F}, \|\cdot\|_\infty\bigr)$.*

Proof (of Lemma 13)
First claim: $\mathcal{F}^{(0)}$ is a γ-net of $\bigl(\mathcal{F}, \|\cdot\|_\infty\bigr)$.
Let $f \in \mathcal{F}$. We explain why there exist $c_1^{(0)}, \dots, c_{1/\gamma}^{(0)} \in C^{(0)}$ such that

$$f^{(0)}(x) = \sum_{a=1}^{1/\gamma} c_a^{(0)} \mathbb{I}_{x \in I_a}$$

satisfies $\bigl|f(x) - f^{(0)}(x)\bigr| \leqslant \gamma$ for all $x \in [0,1]$. We can choose $c_a^{(0)} \in \operatorname{argmin}_{c \in C^{(0)}} \bigl|f(x_a) - c\bigr|$, where x_a is the center of the subinterval I_a. Indeed, since we can approximate $f(x_a)$ with precision $\gamma/2$ (the y-axis discretization is of width γ), and since f is 1-Lipschitz on I_a, we have that, for all $a \in \{1, \dots, 1/\gamma\}$ and all $x \in I_a$,

$$\bigl|f(x) - c_a^{(0)}\bigr| \leqslant \bigl|f(x) - f(x_a)\bigr| + \bigl|f(x_a) - c_a^{(0)}\bigr| \leqslant \frac{\gamma}{2} + \frac{\gamma}{2} = \gamma \,.$$

Since the subintervals I_a, $a \leqslant 1/\gamma$, form a partition of $[0,1]$, we just showed that $\|f - f^{(0)}\|_\infty \leqslant \gamma$.

Second claim: $\mathcal{F}^{(M)}$ is a $\gamma/2^{M+1}$-net of $\bigl(\mathcal{F}, \|\cdot\|_\infty\bigr)$.
Let $f \in \mathcal{F}$. We explain why there exist constants $c_a^{(0)} \in C^{(0)}$ and $c_a^{(m,n)} \in \bigl[-\gamma/2^{m-1}, \gamma/2^{m-1}\bigr]$ such that

$$f_{\boldsymbol{c}}(x) = \sum_{a=1}^{1/\gamma} c_a^{(0)} \mathbb{I}_{x \in I_a} + \sum_{m=1}^{M} \sum_{a=1}^{1/\gamma} \sum_{n=1}^{2^m} c_a^{(m,n)} \mathbb{I}_{x \in I_a^{(m,n)}}$$

satisfies $\bigl|f(x) - f_{\boldsymbol{c}}(x)\bigr| \leqslant \gamma/2^{M+1}$ for all $x \in [0,1]$. We argue below that it suffices to:

- choose the constants $c_a^{(0)} \in \operatorname{argmin}_{c \in C^{(0)}} \bigl|f(x_a) - c\bigr|$ exactly as for $\mathcal{F}^{(0)}$ above;
- choose the constants $c_a^{(m,n)}$ in such a way that, for all levels $m \in \{1, \dots, M\}$, and for all positions $a \in \{1, \dots, 1/\gamma\}$ and $n \in \{1, \dots, 2^m\}$,

$$f\bigl(x_a^{(m,n)}\bigr) = c_a^{(0)} + \sum_{m'=1}^{m} c_a^{(m', n_{m'})} \,, \tag{32}$$

  where $x_a^{(m,n)}$ denotes the center of the subinterval $I_a^{(m,n)}$, and where $n_{m'}$ is the unique integer n' such that $I_a^{(m,n)} \subseteq I_a^{(m',n')}$. Such a choice can be done in a recursive way (induction on m). It is feasible since the functions in $\mathcal{F}$ are 1-Lipschitz (see Figure 1 for an illustration).

To conclude, it is now sufficient to use (32) with $m = M$. Note indeed from (17) that, on each level-M subinterval $I_a^{(M,n)}$, the function $f_{\boldsymbol{c}}$ is equal to

$$f_{\boldsymbol{c}}(x) = c_a^{(0)} + \sum_{m=1}^{M} c_a^{(m,n_m)} \,,$$

where n_m is the unique integer n' such that $I_a^{(M,n)} \subseteq I_a^{(m,n')}$. Thus, by (32), we can see that $f_{\boldsymbol{c}}\big(x_a^{(M,n)}\big) = f\big(x_a^{(M,n)}\big)$ for all points $x_a^{(M,n)}$, $a \in \{1,\ldots,1/\gamma\}$ and $n \in \{1,\ldots,2^M\}$.

Now, if $x \in I_a^{(M,n)}$ is any point in $I_a^{(M,n)}$, then it is at most at a distance of $\gamma/2^{M+1}$ of the middle point $x_a^{(M,n)}$. Therefore, by 1-Lipschitzity of f, we have $\big|f(x) - f\big(x_a^{(M,n)}\big)\big| \leqslant \gamma/2^{M+1}$. Using the equality $f_{\boldsymbol{c}}\big(x_a^{(M,n)}\big) = f\big(x_a^{(M,n)}\big)$ proved above and the fact that $f_{\boldsymbol{c}}$ is constant on $I_a^{(M,n)}$, we get that

$$\forall a \in \{1,\ldots,1/\gamma\}, \quad \forall n \in \{1,\ldots,2^M\}, \quad \forall x \in I_a^{(M,n)}\,, \qquad \big|f(x) - f_{\boldsymbol{c}}(x)\big| \leqslant \gamma/2^{M+1}\,.$$

Since the level-M subintervals $I_a^{(M,n)}$, $a \in \{1,\ldots,1/\gamma\}$ and $n \in \{1,\ldots,2^M\}$, form a partition of $[0,1]$, we just showed that $\|f - f_{\boldsymbol{c}}\|_\infty \leqslant \gamma/2^{M+1}$, which concludes the proof. ■

D.3. An efficient γ-net for Hölder classes (proof of Lemma 10)

First claim: $\mathcal{F}^{(0)}$ is a γ-net of $\big(\mathcal{F}, \|\cdot\|_\infty\big)$.
Let $f \in \mathcal{F}$. We explain why there exist $P_a^{(0)} \in \mathcal{P}_a^{(0)}$ for all $a \in \{1,\ldots,1/\delta_x\}$ such that

$$f^{(0)}(x) = \sum_{a=1}^{1/\delta_x} P_a^{(0)}(x)\mathbb{I}_{x\in I_a}$$

satisfies $\big|f(x) - f^{(0)}(x)\big| \leqslant \gamma$ for all $x \in [0,1]$. Fix $a \in \{1,\ldots,1/\delta_x\}$ and let x_a be the center of the subinterval I_a. By Taylor's formula for all $x \in I_a$ there exist $\xi \in I_a$ such that

$$f(x) = f(x_a) + f'(x_a)(x-x_a) + \frac{f''(x_a)}{2!}(x-x_a)^2 + \cdots + \frac{f^{(q)}(x_a)}{q!}(x-x_a)^q \\ + \frac{1}{q!}\left(f^{(q)}(\xi) - f^{(q)}(x_a)\right)(x-x_a)^q\,. \qquad (33)$$

Thus, the function f can be written as the sum of a polynomial and a term (the last one) that will be proven to be small by the Hölder property (22). Now, for every derivative $i \in \{0,\ldots,q\}$ we can choose $b_i \in \mathcal{Y}^{(0)}$ such that

$$|f^{(i)}(x_a) - b_i| \leqslant \delta_y/2\,. \qquad (34)$$

Indeed, the y-axis discretization $\mathcal{Y}^{(0)}$ of $[-B,B]$ is of width δ_y and $|f^{(i)}(x_a)| \leqslant B$ by definition of $\mathcal{F}$. Thus, setting

$$P_a^{(0)}(x) = b_0 + \frac{b_1}{1}(x-x_a) + \frac{b_2}{2!}(x-x_a)^2 + \cdots + \frac{b_q}{q!}(x-x_a)^q\,,$$

the polynomial $P_a^{(0)}$ satisfies by (33) for all $x \in I_a$

$$\begin{aligned}\big|f(x) - P_a^{(0)}(x)\big| &\leqslant \sum_{i=0}^{q} \frac{\big|f^{(i)}(x_a) - b_i\big|}{i!}|x-x_a|^i + \frac{1}{q!}\left|f^{(q)}(\xi) - f^{(q)}(x_a)\right| |x-x_a|^q \\ &\leqslant \sum_{i=0}^{q} \frac{\delta_y}{2i!} \underbrace{|x-x_a|^i}_{\leqslant 1} + \frac{\lambda}{q!}\,|\xi - x_a|^\alpha\,|x-x_a|^q\,,\end{aligned}$$

where the second inequality is by (34) and because $f^{(q)}$ is α-Hölder with coefficient λ. Now, since $|\xi - x_a|$ and $|x - x_a|$ are bounded by $\delta_x/2$, it yields

$$\big|f(x) - P_a^{(0)}(x)\big| \leqslant \sum_{i=0}^{q} \frac{\delta_y}{2i!} + \frac{\lambda}{q!}\left(\frac{\delta_x}{2}\right)^{q+\alpha} \leqslant \frac{e}{2}\delta_y + \frac{\lambda}{q!}\left(\frac{\delta_x}{2}\right)^{\beta}\,.$$

The choices $\delta_x = 2(q!\gamma/(2\lambda))^{1/\beta}$ and $\delta_y = \gamma/e$ finally entail

$$\left|f(x) - \big[P_a^{(0)}(x)\big]_B\right| \leqslant \big|f(x) - P_a^{(0)}(x)\big| \leqslant \frac{\gamma}{2} + \frac{\gamma}{2} = \gamma\,.$$

This concludes the first part of the proof.

Second claim: $\mathcal{F}^{(M)}$ *is a* $\gamma/2^M$*-net of* $\big(\mathcal{F}, \|\cdot\|_\infty\big)$.
Let $f \in \mathcal{F}$. We explain why there exist clipped-polynomials $P_a^{(0)} \in \mathcal{P}^{(0)}$ and $Q_a^{(m,n)} \in \mathcal{Q}_a^{(m,n)}$ such that

$$f_{\boldsymbol{c}}(x) = \sum_{a=1}^{1/\delta_x} P_a^{(0)}(x)\mathbb{I}_{x\in I_a} + \sum_{m=1}^{M}\sum_{a=1}^{1/\delta_x}\sum_{n=1}^{4^m} Q_a^{(m,n)}(x)\mathbb{I}_{x\in I_a^{(m,n)}}$$

satisfies $\big|f(x) - f_{\boldsymbol{c}}(x)\big| \leqslant \gamma/2^M$ for all $x \in [0,1]$. To do so, we show first that there exist clipped polynomials $P_a^{(0)} \in \mathcal{P}^{(0)}$ and $P_a^{(m,n)} \in \mathcal{P}_a^{(m,n)}$ such that

$$\begin{aligned}\tilde{f}_{\boldsymbol{c}}(x) = \sum_{a=1}^{1/\delta_x} P_a^{(0)}(x)\mathbb{I}_{x\in I_a} &+ \sum_{n=1}^{4}\sum_{a=1}^{1/\delta_x}\big(P_a^{(1,n)} - P_a^{(0)}\big)(x)\mathbb{I}_{x\in I_a^{(1,n)}} \\ &+ \sum_{m=2}^{M}\sum_{a=1}^{1/\delta_x}\sum_{n=1}^{4^m}\Big(P_a^{(m,n)} - P_a^{(m-1,n_{m-1})}\Big)(x)\mathbb{I}_{x\in I_a^{(m,n)}}\end{aligned}$$

satisfies $\big|f(x) - \tilde{f}_{\boldsymbol{c}}(x)\big| \leqslant \gamma/2^M$ for all $x \in [0,1]$. We recall that n_{m-1} denotes the unique integer n' such that $I_a^{(m,n)} \subset I_a^{(m-1,n')}$. First we remark that the function $\tilde{f}_c$ defined above equals $P_a^{(M,n)}$ on each level-M subinterval $I_a^{(M,n)}$.

Thus, it suffices to design clipped polynomials $P_a^{(m,n)} \in \mathcal{P}_a^{(m,n)}$, such that $\big|f(x) - P_a^{(m,n)}(x)\big| \leqslant \gamma/2^m$ for all $x \in I_a^{(m,n)}$. To do so, we reproduce the same proof as for $\mathcal{F}^{(0)}$ above. Because $\operatorname{diam} I_a^{(m,n)} = \delta_x/4^m \leqslant \delta_x/2^{m/\beta}$ (recall that $\beta \geqslant 1/2$), for every position $a \in \{1,\dots,1/\delta_x\}$, every level $m \in \{1, \ldots, M\}$, and every $n \in \{1, \ldots, 4^m\}$, we can define as for $\mathcal{F}^{(0)}$ above a polynomial

$$\tilde{P}_a^{(m,n)}(x) = b_0 + \frac{b_1}{1}\left(x - x_a^{(m,n)}\right) + \frac{b_2}{2!}\left(x - x_a^{(m,n)}\right)^2 + \cdots + \frac{b_q}{q!}\left(x - x_a^{(m,n)}\right)^q$$

(recall that $x_a^{(m,n)}$ is the center of $I_a^{(m,n)}$) such that all coefficients b_j have the form $-B + z_j \delta_y 2^{-m}$ for some $z_j \in \{0, \ldots, 2^{m+1} B/\delta_y\}$ and

$$\left| f(x) - \left[\tilde{P}_a^{(m,n)}(x)\right]_B \right| \leqslant \gamma/2^m \tag{35}$$

for all $x \in I_a^{(m,n)}$. To conclude, we choose the clipped polynomials $P_a^{(m,n)} = \left[\tilde{P}_a^{(m,n)}\right]_B$.

To conclude the proof, we see that for all $x \in I_a^{(m,n)}$, by the triangle inequality

$$\left| P_a^{(m,n)}(x) - P_a^{(m-1,n_{m-1})}(x) \right| \leqslant \frac{\gamma}{2^m} + \frac{\gamma}{2^{m-1}} = \frac{3\gamma}{2^m}\,,$$

so that $f_c = \tilde{f}_c$ for the choices $Q_a^{(m,n)} = \left[P_a^{(m,n)}(x) - P_a^{(m-1,n_{m-1})}(x) \right]_{3\gamma/2^m}$.

D.4. Proof of Theorem 6

We split our proof into two main parts. First, we explain why each functions $\widehat{f}_{t,a}$ incurs small cumulative regret inside each subinterval I_a. Second, we sum the previous regret bounds over all positions $a \in \{1, \ldots, 1/\gamma\}$.

Part 1: focus on a subinterval I_a In this part, we fix some $a \in \{1, \ldots, 1/\gamma\}$ and we consider the a-th instance of the algorithm $\mathcal{A}$, whose local time is only incremented when a new x_t falls into I_a. As in Algorithm 2, our instance of algorithm $\mathcal{A}$ uses a combination of the EWA and the Multi-variable EG forecasters to perform high-scale and low-scale aggregation simultaneously. Thus, the proof closely follows the path of the one of Theorem 2. We split again the proof into two subparts: one for each level of aggregation.

Subpart 1: low-scale aggregation.

In this subpart, we fix $j \in \{0, \ldots, 2B/\gamma\}$. The proof starts as the one of Theorem 2 except that $\mathcal{A}$ applies the adaptive version of the Multi-variable Exponentiated Gradient forecaster (Algorithm 3, Appendix B) with the loss function ℓ_t defined in (18). We will thus apply Theorem 9 (available in Appendix B) instead of Theorem 1. After checking its assumptions exactly as in the proof of Theorem 2, we can apply Theorem 9. The norms of the loss gradients $\left\|\nabla_{\widehat{u}_t^{(m,n)}} \ell_t\right\|_\infty$ are bounded by $16B\gamma/2^m$ if x_t falls in $I_a^{(m,n)}$ and by 0 otherwise. Setting $T_a^{(m,n)} = \sum_{t=1}^T \mathbb{I}_{x_t \in I_a^{(m,n)}}$, Theorem 9 yields as in (11):

$$\sum_{t=1}^T \left(y_t - \widehat{f}_{t,a,j}(x_t) \right)^2 \mathbb{I}_{x_t \in I_a} \tag{36}$$

$$\leqslant \inf_{c_a^{(m,n)},\, \forall (m,n)} \sum_{t=1}^{T} \left(y_t - \left(-B + j\gamma + \sum_{m=1}^{M} \sum_{n=1}^{2^m} c_a^{(m,n)} \mathbb{I}_{x_t \in I_a^{(m,n)}} \right) \right)^2 \mathbb{I}_{x_t \in I_a} + 2 \sum_{m=1}^{M} \sum_{n=1}^{2^m} 16 B\gamma / 2^m \sqrt{T_a^{(m,n)} \log 2}\,, \tag{37}$$

where the infimum is over all constants $c_a^{(m,n)} \in [-\gamma/2^{m-1}, \gamma/2^{m-1}]$ for every $m = 1, \ldots, M$ and $n = 1, \ldots, 2^m$. But, for each level $m = 1, \ldots, M$, the point x_t only falls into one interval $I_a^{(m,n)}$. Thus, $\sum_{n=1}^{2^m} T_a^{(m,n)} = T_a$, where $T_a = \sum_{t=1}^{T} \mathbb{I}_{x_t \in I_a}$ is the final local time of the a-th instance of $\mathcal{A}$. Therefore, using the concavity of the square root and applying Jensen's inequality, (37) entails

$$\begin{aligned}
&\sum_{t=1}^{T} \left(y_t - \widehat{f}_{t,a,j}(x_t) \right)^2 \mathbb{I}_{x_t \in I_a} \\
&\quad\leqslant \inf_{c_a^{(m,n)},\, \forall (m,n)} \sum_{t=1}^{T} \left(y_t - \left(-B + j\gamma + \sum_{(m,n)} c_a^{(m,n)} \mathbb{I}_{x_t \in I_a^{(m,n)}} \right) \right)^2 \mathbb{I}_{x_t \in I_a} \\
&\qquad\qquad + 32 B\gamma \sum_{m=1}^{M} 2^{-m} \sqrt{T_a 2^m \log 2} \\
&\quad\leqslant \inf_{c_a^{(m,n)},\, \forall (m,n)} \sum_{t=1}^{T} \left(y_t - \left(-B + j\gamma + \sum_{(m,n)} c_a^{(m,n)} \mathbb{I}_{x_t \in I_a^{(m,n)}} \right) \right)^2 \mathbb{I}_{x_t \in I_a} \\
&\qquad\qquad + 32 B\gamma (1 + \sqrt{2}) \sqrt{T_a \log 2}\,.
\end{aligned} \tag{38}$$

The second inequality is because $\sum_{m=1}^{\infty} 2^{-m/2} = 1 + \sqrt{2}$.

Subpart 2: high-scale aggregation.

Following the proof of Theorem 2, we apply EWA to the experts $\widehat{f}_{t,a,j}$ for $j \in \{0, \ldots, 2B/\gamma\}$ with tuning parameter $\eta = 1/(2(4B)^2)$ because $\widehat{f}_{t,a,j} \in [-B - 2\gamma, B + 2\gamma] \subset [-3B, 3B]$ and $y_t \in [-B, B]$. We get from Proposition 3.1 and Page 46 of Cesa-Bianchi and Lugosi (2006) that

$$\begin{aligned}
&\sum_{t=1}^{T} \left(y_t - \widehat{f}_{t,a}(x_t) \right)^2 \mathbb{I}_{x_t \in I_a} \\
&\quad\leqslant \min_{0 \leqslant j \leqslant 2B/\gamma} \sum_{t=1}^{T} \left(y_t - \widehat{f}_{t,a,j}(x_t) \right)^2 \mathbb{I}_{x_t \in I_a} + \frac{\log\left(2B/\gamma + 1\right)}{\eta} \\
&\quad\leqslant \min_{0 \leqslant j \leqslant 2B/\gamma} \inf_{c_a^{(m,n)},\, \forall (m,n)} \sum_{t=1}^{T} \left(y_t - \left(-B + j\gamma + \sum_{(m,n)} c_a^{(m,n)} \mathbb{I}_{x \in I_a^{(m,n)}} \right) \right)^2 \mathbb{I}_{x_t \in I_a} \\
&\qquad\qquad + 32 B \big(1 + \sqrt{2}\big) \gamma \sqrt{T_a \log 2} \;+ 32 B^2 \log\left(2B/\gamma + 1\right)\;,
\end{aligned} \tag{39}$$

where the infima are over all $j \in \{0, \ldots, 2B/\gamma\}$ and all constants $c_a^{(m,n)} \in [-\gamma/2^{m-1}, \gamma/2^{m-1}]$, and where the second inequality follows from (38) and from $\eta = 1/(32B^2)$.

Part 2: we sum the regrets over all subintervals I_a By definition of $\widehat{f}_t$, we have

$$\sum_{t=1}^{T} \big(y_t - \widehat{f}_t(x_t)\big)^2 = \sum_{t=1}^{T} \left(y_t - \sum_{a=1}^{1/\gamma} \widehat{f}_{t,a}(x_t)\mathbb{I}_{x_t \in I_a}\right)^2$$
$$= \sum_{a=1}^{1/\gamma}\sum_{t=1}^{T} \left(y_t - \widehat{f}_{t,a}(x_t)\right)^2 \mathbb{I}_{x_t \in I_a}$$

Now, by definition of $\mathcal{F}^{(M)}$, summing (39) over all $a = 1, \dots, 1/\gamma$ leads to

$$\sum_{t=1}^{T} \big(y_t - \widehat{f}_t(x_t)\big)^2 \leqslant \inf_{f \in \mathcal{F}^{(M)}} \sum_{t=1}^{T} \big(y_t - f(x_t)\big)^2 + \frac{32B^2}{\gamma}\log\left(2B/\gamma + 1\right)$$
$$+ 32B\big(1+\sqrt{2}\big)\gamma\sqrt{\log 2}\left(\sum_{a=1}^{1/\gamma}\sqrt{T_a}\right) . \qquad (40)$$

Then, using that $\sum_{a=1}^{1/\gamma} T_a = T$, since at every round t, the point x_t only falls into one subinterval I_a, and applying Jensen's inequality to the square root, we can see that

$$\sum_{a=1}^{1/\gamma}\sqrt{T_a} \leqslant \sqrt{T/\gamma}\,.$$

Therefore, substituting in (40), we obtain

$$\sum_{t=1}^{T} \big(y_t - \widehat{f}_t(x_t)\big)^2 \leqslant \inf_{f \in \mathcal{F}^{(M)}} \sum_{t=1}^{T} \big(y_t - f(x_t)\big)^2 + \frac{32B^2}{\gamma}\log\left(2B/\gamma + 1\right)$$
$$+ 32B\big(1+\sqrt{2}\big)\sqrt{\gamma T \log 2}\ . \qquad (41)$$

But, $\mathcal{F}^{(M)}$ is by Lemma 13 a $\gamma/2^{M+1}$-net of $\mathcal{F}$. Using that $M = \lceil \log_2(\gamma T/B) \rceil$ and following the proof of (15), it entails

$$\inf_{f \in \mathcal{F}^{(M)}} \sum_{t=1}^{T} \big(y_t - f(x_t)\big)^2 \leqslant \inf_{f \in \mathcal{F}} \sum_{t=1}^{T} \big(y_t - f(x_t)\big)^2 + 2B^2 + \frac{B^2}{4T}\,.$$

Finally, from (41) we have

$$\sum_{t=1}^{T} \big(y_t - \widehat{f}_t(x_t)\big)^2 \leqslant \inf_{f \in \mathcal{F}} \sum_{t=1}^{T} \big(y_t - f(x_t)\big)^2 + \frac{32B^2}{\gamma}\log\left(2B/\gamma + 1\right)$$
$$+ 32B\big(1+\sqrt{2}\big)\sqrt{\gamma T \log 2} + 2B^2 + \frac{B^2}{4T}\ . \qquad (42)$$

The above regret bound grows roughly as (we omit logarithmic factors and small additive terms):

$$\gamma^{-1} + \sqrt{\gamma T}\,.$$

Optimizing in γ would yield $\gamma \approx T^{-1/3}$ and a regret roughly of the order of $T^{1/3}$. More rigorously, taking $\gamma = BT^{-1/3}$ and substituting it in (42) concludes the proof.

D.5. Proof of Theorem 11

The proof closely follows the one of Theorem 6. It is split into two main parts. First, we explain why each function $\widehat{f}_{t,a}$ incurs a small cumulative regret inside each subinterval I_a. Second, we sum the previous regret bounds over all positions $a = 1, \dots, 1/\delta_x$.

Part 1: focus on a subinterval I_a In this part, we fix some $a \in \{1, \dots, 1/\delta_x\}$ and we consider the a-th instance of the algorithm $\mathcal{A}$, denoted $\mathcal{A}_a$, whose local time is only incremented when a new x_t falls into I_a. As in Algorithm 2, $\mathcal{A}_a$ uses a combination of the EWA and the Multi-variable EG forecasters to perform high-scale and low-scale aggregation simultaneously. We split again the proof into two subparts: one for each level of aggregation.

Subpart 1: low-scale aggregation.

In this subpart, we fix $j \in \{1, \dots, \operatorname{card} \mathcal{P}_a^{(0)}\}$. Similarly to the proof of Theorem 6, we start by applying Theorem 9. Since the elements in $\mathcal{Q}_a^{(m,n)}$ are bounded in supremum norm by $3\gamma/2^m$, and since the elements in $\mathcal{P}_a^{(0)}$ are bounded by B, the norms of the gradients of the loss function (defined in (27)) are bounded by 0 if $x_t \notin I_a^{(m,n)}$ and as follows otherwise:

$$\big\|\nabla_{\widehat{u}^{(m,n)}_{t,a,j}} \ell_t\big\|_\infty \leqslant 2\Big(|y_t| + \big\|\widehat{f}_{t,a,j}\big\|_\infty\Big)\Big\|Q^{(m,n)}_{a,k}\Big\|_\infty \leqslant 2(B+4B)3\gamma/2^m = 30B\gamma/2^m\,.$$

Here, we used that

$$\big|\widehat{f}_{t,a,j}(x)\big| \leqslant \big\|P^{(0)}_{a,j}\big\|_\infty + \sum_{m=1}^{M}\sum_{n=1}^{4^m}\sum_{k=1}^{\operatorname{card}\mathcal{Q}_a^{(m,n)}} \widehat{u}^{(m,n)}_{t,a,j,k}\Big|Q^{(m,n)}_{a,k}(x)\Big|\mathbb{I}_{x\in I_a^{(m,n)}} \leqslant B + \sum_{m=1}^{M} \frac{3\gamma}{2^m} \leqslant 4B\,. \tag{43}$$

Thus, setting $T_a^{(m,n)} = \sum_{t=1}^T \mathbb{I}_{x_t \in I_a^{(m,n)}}$, Theorem 9 yields:

$$\begin{aligned}
&\sum_{t=1}^{T}\Big(y_t - \widehat{f}_{t,a,j}(x_t)\Big)^2\mathbb{I}_{x_t\in I_a} \\
&\quad\leqslant \inf_{Q_a^{(m,n)},\ \forall (m,n)} \sum_{t=1}^{T}\left(y_t - \left(P^{(0)}_{a,j} + \sum_{m=1}^{M}\sum_{n=1}^{4^m} Q_a^{(m,n)}\mathbb{I}_{x_t\in I_a^{(m,n)}}\right)(x_t)\right)^2\mathbb{I}_{x_t\in I_a} \\
&\qquad\qquad + 2\sum_{m=1}^{M}\sum_{n=1}^{4^m} 30B\gamma/2^m\sqrt{T_a^{(m,n)}\log\Big(\operatorname{card}\mathcal{Q}_a^{(m,n)}\Big)}\,,
\end{aligned} \tag{44}$$

where the infimum is over all polynomial functions $Q_a^{(m,n)} \in \mathcal{Q}_a^{(m,n)}$ for every $m = 1, \dots, M$ and $n = 1, \dots, 4^m$. But, for each level $m = 1, \dots, M$, the point x_t only falls into one interval $I_a^{(m,n)}$. Thus, $\sum_{n=1}^{4^m} T_a^{(m,n)} = T_a$, where $T_a = \sum_{t=1}^T \mathbb{I}_{x_t\in I_a}$ is the final local time of the a-th instance of $\mathcal{A}$. Therefore, using the concavity of the square root and applying Jensen's inequality, (44) entails

$$\sum_{t=1}^{T}\Big(y_t - \widehat{f}_{t,a,j}(x_t)\Big)^2\mathbb{I}_{x_t\in I_a} \tag{45}$$

$$\leqslant \inf_{Q_a^{(m,n)},\ \forall (m,n)} \sum_{t=1}^{T} \left(y_t - \left(P_{a,j}^{(0)} + \sum_{(m,n)} Q_a^{(m,n)} \mathbb{I}_{x_t \in I_a^{(m,n)}} \right)(x_t) \right)^2 \mathbb{I}_{x_t \in I_a}$$

$$+ 60B\gamma \sum_{m=1}^{M} 2^{-m} \sqrt{T_a 4^m \log\left(\operatorname{card} \mathcal{Q}_a^{(m,n)} \right)}\,. \tag{46}$$

Now, by the definitions of $\mathcal{Q}_a^{(m,n)}$, $\mathcal{P}_a^{(m,n)}$, and $\mathcal{Y}^{(m)}$ (see Equations (24) and (25)), we can see that

$$\operatorname{card} \mathcal{Q}_a^{(m,n)} \leqslant \operatorname{card} \left(\mathcal{P}_a^{(m,n)} \right)^2 \leqslant \left(\operatorname{card} \mathcal{Y}^{(m)} \right)^{2(q+1)} = \left(2^{m+1} B/\delta_y + 1 \right)^{2(q+1)}$$
$$= \left(2^{m+1} eB/\gamma + 1 \right)^{2(q+1)},$$

which yields

$$\sum_{m=1}^{M} 2^{-m} \sqrt{4^m \log\left(\operatorname{card} \mathcal{Q}_a^{(m,n)} \right)} \leqslant \sum_{m=1}^{M} \sqrt{2(q+1) \log\left(2^{m+1} eB/\gamma + 1 \right)}$$
$$\leqslant M \sqrt{2(q+1) \log\left(2^{M+1} eB/\gamma + 1 \right)}\,.$$

Thus, using $M = \lceil \log_2(\gamma T/B) \rceil$, so that $2^M \gamma^{-1} \leqslant 2T/B$ and combining the above inequality with (46), we have

$$\sum_{t=1}^{T} \left(y_t - \widehat{f}_{t,a,j}(x_t) \right)^2 \mathbb{I}_{x_t \in I_a} \tag{47}$$

$$\leqslant \inf_{Q_a^{(m,n)},\ \forall (m,n)} \sum_{t=1}^{T} \left(y_t - \left(P_{a,j}^{(0)} + \sum_{(m,n)} Q_a^{(m,n)} \mathbb{I}_{x_t \in I_a^{(m,n)}} \right)(x_t) \right)^2 \mathbb{I}_{x_t \in I_a}$$

$$+ 60B\gamma \lceil \log_2(\gamma T/B) \rceil \sqrt{2(q+1) T_a \log(4eT + 1)}\,. \tag{48}$$

Subpart 2: high-scale aggregation.

Following the proof of Theorem 6, we apply EWA to the experts $\widehat{f}_{t,a,j}$ for $j \in \left\{ 1, \ldots, \operatorname{card} \mathcal{P}_a^{(0)} \right\}$ with tuning parameter $\eta = 1/(2(5B)^2) = 1/(50B^2)$ because $\widehat{f}_{t,a,j} \in [-4B, 4B]$ (see (43)). From (48) and using $\operatorname{card} \mathcal{P}_a^{(0)} \leqslant (2B/\delta_y + 1)^{q+1} = (2eB/\gamma + 1)^{q+1}$, we have

$$\sum_{t=1}^{T} \left(y_t - \widehat{f}_{t,a}(x_t) \right)^2 \mathbb{I}_{x_t \in I_a}$$

$$\leqslant \min_{1 \leqslant j \leqslant \operatorname{card} \mathcal{P}_a^{(0)}} \sum_{t=1}^{T} \left(y_t - \widehat{f}_{t,a,j}(x_t) \right)^2 \mathbb{I}_{x_t \in I_a} + \frac{\log\left(\operatorname{card} \mathcal{P}_a^{(0)} \right)}{\eta}$$

$$
\begin{aligned}
&\leqslant \inf_{P_a^{(0)} \in \mathcal{P}_a^{(0)}} \quad \inf_{Q_a^{(m,n)},\, \forall (m,n)} \quad \sum_{t=1}^{T} \left(y_t - \left(P_a^{(0)} + \sum_{(m,n)} Q_a^{(m,n)} \mathbb{I}_{x_t \in I_a^{(m,n)}} \right)(x_t) \right)^2 \mathbb{I}_{x_t \in I_a} \\
&\qquad + 60 B\gamma \lceil \log_2(\gamma T/B) \rceil \sqrt{2(q+1) T_a \log(4eT+1)} \\
&\qquad + 50 B^2 (q+1) \log\left(2eB/\gamma + 1\right) \,, \qquad (49)
\end{aligned}
$$

where the infimum is over all functions $P^{(0)} \in \mathcal{P}_a^{(0)}$ and $Q_a^{(m,n)} \in \mathcal{Q}_a^{(m,n)}$, and where the second inequality follows from $\eta = 1/(50B^2)$.

Part 2: we sum the regrets over all subintervals I_a By definition of $\widehat{f}_t$, we have

$$
\begin{aligned}
\sum_{t=1}^{T} \left(y_t - \widehat{f}_t(x_t)\right)^2 &= \sum_{t=1}^{T} \left(y_t - \sum_{a=1}^{1/\delta_x} \widehat{f}_{t,a}(x_t) \mathbb{I}_{x_t \in I_a} \right)^2 \\
&= \sum_{a=1}^{1/\delta_x} \sum_{t=1}^{T} \left(y_t - \widehat{f}_{t,a}(x_t) \right)^2 \mathbb{I}_{x_t \in I_a}
\end{aligned}
$$

Now, by definition of $\mathcal{F}^{(M)}$, summing (49) over all $a = 1, \dots, 1/\delta_x$ leads to

$$
\begin{aligned}
\sum_{t=1}^{T} \left(y_t - \widehat{f}_t(x_t)\right)^2 \leqslant \inf_{f \in \mathcal{F}^{(M)}} &\sum_{t=1}^{T} \left(y_t - f(x_t)\right)^2 \\
&+ 50B^2(q+1)\log(2eB/\gamma+1)\delta_x^{-1} \\
&+ 60B\gamma \lceil \log_2(\gamma T/B) \rceil \sqrt{2(q+1)\log(4eT+1)} \left(\sum_{a=1}^{1/\delta_x} \sqrt{T_a} \right) .
\end{aligned}
\qquad (50)
$$

Then, using that $\sum_{a=1}^{1/\delta_x} T_a = T$, since at every round t, the point x_t only falls into one subinterval I_a, and applying Jensen's inequality to the square root, we can see that

$$
\sum_{a=1}^{1/\delta_x} \sqrt{T_a} \leqslant \sqrt{T/\delta_x}\,.
$$

Therefore, substituting in (50) and because $\delta_x = 2(q!\gamma/(2\lambda))^{1/\beta}$, we have

$$
\begin{aligned}
\sum_{t=1}^{T} \left(y_t - \widehat{f}_t(x_t)\right)^2 \leqslant \inf_{f \in \mathcal{F}^{(M)}} &\sum_{t=1}^{T} \left(y_t - f(x_t)\right)^2 \\
&+ 25B^2(q+1)\log(2eB/\gamma+1)\big(q!\gamma/(2\lambda)\big)^{-1/\beta} \\
&+ 60B\gamma \lceil \log_2(\gamma T/B) \rceil \sqrt{(q+1)\log(4eT+1) T \big(q!\gamma/(2\lambda)\big)^{-1/\beta}}\,.
\end{aligned}
$$

But, $\mathcal{F}^{(M)}$ is by Lemma 10 a $\gamma/2^M$-net of $\mathcal{F}$. Using that $M = \lceil \log_2(\gamma T/B) \rceil$ and following the proof of (15), it entails

$$
\inf_{f \in \mathcal{F}^{(M)}} \sum_{t=1}^{T} \left(y_t - f(x_t)\right)^2 \leqslant \inf_{f \in \mathcal{F}} \sum_{t=1}^{T} \left(y_t - f(x_t)\right)^2 + 4B^2 + \frac{B^2}{T}\,.
$$

Finally, we have

$$\sum_{t=1}^{T}\big(y_t-\widehat{f}_t(x_t)\big)^2 \leqslant \inf_{f\in\mathcal{F}}\sum_{t=1}^{T}\big(y_t-f(x_t)\big)^2 \\ + 25B^2(q+1)\log(2eB/\gamma+1)\big(q!\gamma/(2\lambda)\big)^{-1/\beta}+4B^2+\frac{B^2}{T} \\ + 60B\gamma\lceil\log_2(\gamma T/B)\rceil\sqrt{(q+1)\log(4eT+1)T\big(q!\gamma/(2\lambda)\big)^{-1/\beta}}\,. \tag{51}$$

The above regret bound grows roughly as (we omit logarithmic factors and small additive terms):

$$\gamma^{-1/\beta}+\gamma^{1-1/(2\beta)}\sqrt{T}\,.$$

Optimizing in γ would yield $\gamma\approx T^{-\beta/(2\beta+1)}$ and a regret roughly of order $\mathcal{O}\big(T^{1/(2\beta+1)}\big)$. More rigorously, taking $\gamma=BT^{-\beta/(2\beta+1)}$ and substituting it in (51) concludes the proof.

D.6. Proof of Lemma 12

Storage complexity. Fix a position $a\in\{1,\ldots,1/\delta_x\}$. At round $t\geqslant 1$, the Nested Chaining Algorithm for Hölder functions needs to store:

- the high-level weights $\widehat{w}_{t,a,j}$ for every $j\in\big\{1,\ldots,\operatorname{card}\mathcal{P}_a^{(0)}\big\}$;
- the low-level weights $\widehat{u}_{t,a,j,k}^{(m,n)}$ for every $j\in\big\{1,\ldots,\operatorname{card}\mathcal{P}_a^{(0)}\big\}$, every $m\in\{1,\ldots,M\}$, every $n\in\{1,\ldots,4^m\}$, and every $k\in\big\{1,\ldots,\operatorname{card}\mathcal{Q}_a^{(m,n)}\big\}$.

The complexity of the ath instance of $\mathcal{A}$ is thus of order

$$\operatorname{card}\mathcal{P}_a^{(0)}\times M\times 4^M\times\operatorname{card}\mathcal{Q}_a^{(M,n)}\,.$$

Now, we bound each of these terms separately. First for $\gamma=BT^{-\beta/(2\beta+1)}$, we have

$$\operatorname{card}\mathcal{P}_a^{(0)}\leqslant(2B/\delta_y+1)^{q+1}=(2eB/\gamma+1)^{q+1}=\big(2eT^{\beta/(2\beta+1)}+1\big)^{q+1} \\ =\mathcal{O}\big(T^{\beta(q+1)/(2\beta+1)}\big)\,,$$

because $\delta_y=e/\gamma$. Second using $M=\lceil\log_2(\gamma T/B)\rceil$, we can see that

$$4^M=\big(2^M\big)^2\leqslant(2\gamma T/B)^2=\big(2T^{1-\beta/(2\beta+1)}\big)^2=\mathcal{O}\big(T^{2-2\beta/(2\beta+1)}\big)\,,$$

and that

$$\operatorname{card}\mathcal{Q}_a^{(M,n)}\leqslant\big(2^{M+1}eB/\gamma+1\big)^{2(q+1)}\leqslant(4eT+1)^{2(q+1)}=\mathcal{O}\big(T^{2(q+1)}\big)\,.$$

Putting all things together the space-complexity of the ath instance of $\mathcal{A}$ is of order

$$\mathcal{O}\big(T^{2q+4+\beta(q-1)/(2\beta+1)}\log T\big)\,.$$

The whole storage complexity of the algorithm is thus of order

$$\mathcal{O}\bigl(T^{2q+4+\beta(q-1)/(2\beta+1)}/\delta_x\bigr) = \mathcal{O}\bigl(T^{2q+4+(\beta(q-1)+1)/(2\beta+1)} \log T\bigr)\,,$$

where we used that $\delta_x = 2(q!\gamma/(2\lambda))^{1/\beta} = \mathcal{O}\bigl(T^{-1/(2\beta+1)}\bigr)$

Time complexity. At round $t \geqslant 1$, x_t only falls into one subinterval I_a and one subinterval $I_a^{(m,n)}$ for each level $m = 1, \dots, M$. It thus needs to update

- the weights $\widehat{w}_{t,a,j}$ for a single position a and for every $j \in \bigl\{1, \dots, \operatorname{card} \mathcal{P}_a^{(0)}\bigr\}$,
- for every level $m = 1, \dots, M$ the weights $\widehat{u}_{t,a,j,k}^{(m,n)}$ for a single position a and a single n, but for all $j \in \bigl\{1, \dots, \operatorname{card} \mathcal{P}_a^{(0)}\bigr\}$ and all $k \in \bigl\{1, \dots, \operatorname{card} \mathcal{Q}_a^{(m,n)}\bigr\}$.

The time-complexity is thus bounded by

$$\mathcal{O}\Bigl(\operatorname{card} \mathcal{P}_a^{(0)} \times M \times \operatorname{card} \mathcal{Q}_a^{(M,n)}\Bigr) = \mathcal{O}\bigl(T^{(q+1)(2+\beta/(2\beta+1))} \log T\bigr)\,.$$