Minimax-optimal and Locally-adaptive Online Nonparametric Regression

Paul Liautaud[1], Pierre Gaillard[2], Olivier Wintenberger[1,3]

[1]Sorbonne Université, CNRS, LPSM, Paris, France ; [2]Université Grenoble Alpes, INRIA, CNRS, Grenoble INP, LJK, Grenoble, France ; [3]Institut Pauli CNRS, Vienna University, Wien, Austria

Setting: online nonparametric regression

Context & data: data arrives **sequentially** as a stream $x_1, \dots, x_t$ and we want to predict each response as follows:

Learning scenario

For each round $t = 1, \dots, T$, the learner or algorithm

- observes an input $x_t \in \mathcal{X} \subset \mathbb{R}^d$
- makes a prediction $\hat{f}_t(x_t) \in \mathbb{R}$ — *Choose $\hat{f}_t$ before observing ℓ_t*
- suffers a loss $\ell_t(\hat{f}_t(x_t))$ and observes gradient of it — *No assumptions on how ℓ_t is generated*
- updates his rule prediction $\hat{f}_t \to \hat{f}_{t+1}$ — *Based on observed gradients*

Goal: given some large (nonparametric) function set $\mathcal{F} \subset \mathbb{R}^{\mathcal{X}}$ we want to minimize the regret against any competitor $f \in \mathcal{F}$

$$\mathrm{Reg}_T(f) = \underbrace{\sum_{t=1}^{T} \ell_t(\hat{f}_t(x_t))}_{\text{our performance}} - \underbrace{\sum_{t=1}^{T} \ell_t(f(x_t))}_{\text{reference performance}}$$

Assumptions:

- (ℓ_t) are general G-Lipschitz convex losses, $G > 0$ known;
- $\mathcal{X} \subset \mathbb{R}^d$ bounded compact subset;
- $\mathcal{F} \subset [-B, B]^{\mathcal{X}}$, $B > 0$ known;
- $\mathcal{F} \subset \mathscr{C}^\alpha(L)$ the set of α-Hölder continuous functions, $L > 0$ and $\alpha \in (0, 1]$ **unknown**.

No stochastic assumption on data (x_t, ℓ_t): $(\hat{f}_t)$ have to perform well on **abitrary** and possibly **adversarial** data.

Main contributions

❶ A **parameter-free** online learning method that leverages a **chaining tree** structure and achieves a regret over α-Hölder continuous functions $\mathscr{C}^\alpha(L)$:

$$\sup_{f \in \mathscr{C}^\alpha(L)} \mathrm{Reg}_T(f) \lesssim GB\sqrt{T} + GL \begin{cases} \sqrt{T}, & \text{if } d < 2\alpha, \\ \log_2 T \sqrt{T}, & \text{if } d = 2\alpha, \\ T^{1-\frac{\alpha}{d}}, & \text{if } d > 2\alpha. \end{cases}$$

❷ An algorithm that **optimally** competes against any **pruning** and adapts to the **local Hölder regularities** of the competitor, achieving for $d = 1, \alpha \in [\frac{1}{2}, 1]$:

$$\sup_{f \in \mathscr{C}^\alpha(L)} \mathrm{Reg}_T(f) \lesssim \inf_{\text{prun}} \left\{ \sqrt{T|\text{prun}|} + \sum_{n \in \text{prun}} 2^{-\alpha \text{level}(n)} L_n(f) \sqrt{|T_n|} \right\},$$

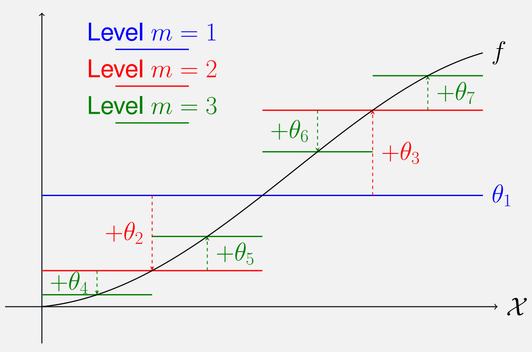
where $(L_n) \leq L$ are local Hölder constants with respect to the pruning nodes.

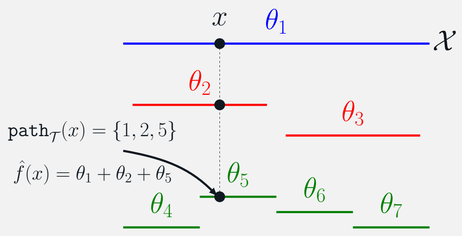
Parameter-free online learning with chaining tree

Chaining tree:

$\mathcal{T} = (\mathcal{X}_n, \theta_n)_{n \in \mathcal{N}(\mathcal{T})}$ associated to predictor

$$\hat{f}(x) = \sum_{n \in \mathcal{N}(\mathcal{T})} \theta_n \mathbb{1}_{\mathcal{X}_n}(x).$$





Multi-scale approach:

$\mathcal{T}$ predicts with all nodes: allows a **multi-scale** prediction and approximation process.

Coefficient decay:

At level $m \geq 1$, given the regularity of $\mathscr{C}^\alpha(L)$,

$$|\theta_{\text{level } m}| \lesssim L 2^{-\alpha m}$$

Algorithm:

Algorithm 1: Online training of chaining tree $\hat{f}_t \to \hat{f}_{t+1}$

Input : Nodes $(\theta_{n,t})$ of $\mathcal{T}$ and gradients $(g_{n,t})$

for $n \in path_{\mathcal{T}}(x_t)$ **do**

Predict $\hat{f}_t(x_t) = \sum_{n \in \mathcal{N}(\mathcal{T})} \theta_n \mathbb{1}_{\mathcal{X}_n}(x_t)$;

Find $\theta_{n,t+1} \in \mathbb{R}$ to approximately minimize

$$\theta_n \mapsto \ell_t(\hat{f}_{-n,t}(x_t) + \theta_n \mathbb{1}_{\mathcal{X}_n}(x_t)) \quad \text{with} \quad \hat{f}_{-n,t}(x_t) = \hat{f}_t(x_t) - \theta_{n,t} \mathbb{1}_{\mathcal{X}_n}(x_t) \qquad (1)$$

using gradient $g_{n,t} = \left[\partial_{\theta_n} \ell_t\big(\hat{f}_{-n,t}(x_t) + \theta_n \mathbb{1}_{\mathcal{X}_n}(x_t)\big) \right]_{\theta_n = \theta_{n,t}} = \ell_t'(\hat{f}_t(x_t))$.

Output: Nodes $(\theta_{n,t+1})$

Parameter-free [3] subroutine in (1) achieves regret based on the norm of the (θ_n)

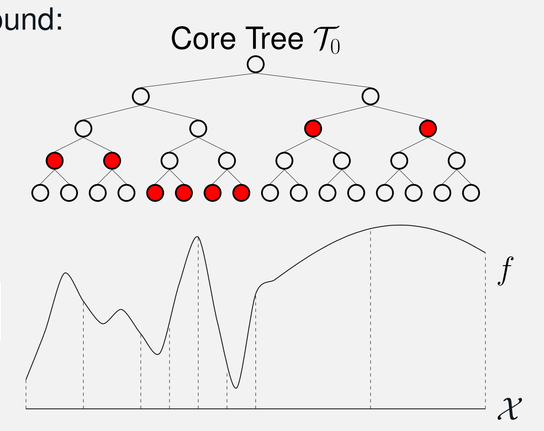
$$G \sum_n |\theta_n| \sqrt{|T_n|} \lesssim GL \sum_m 2^{-\alpha m} \sqrt{2^{dm} T}$$

with $T_n = \{1 \leq t \leq T : x_t \in \mathcal{X}_n\}$ and $|\{\text{nodes at level } m\}| = 2^{d(m-1)}$.

Minimax-optimal and locally-adaptive algorithm

Local-adaptivity: our algorithm leverages expert aggregation procedure and aims to fit the best **pruning** corresponding to **local Hölder profile** of the competitor: we shift from a global to a **local** regret bound:

$$L(f)\sqrt{T} \longrightarrow \sum_{n \in \text{prun}} L_n(f) \sqrt{|T_n|}.$$



Curvature adaptivity: in case (ℓ_t) are exp-concave, for any $f \in \mathscr{C}^\alpha(L)$ and any pruning our algorithm achieves better regret

$$O\big(|\text{prun}| + \sum_{n \in \text{prun}} 2^{-\alpha \text{level}(n)} L_n(f) \sqrt{|T_n|}\big),$$

with $(L_n(f)) \leq L$ local Hölder constants with respect to the pruning nodes.

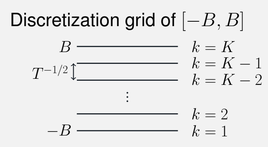
Corollary & minimax-optimality: for $d = 1, \alpha \in [\frac{1}{2}, 1]$ we obtain

$$\sup_{f \in \mathscr{C}^\alpha(L)} \mathrm{Reg}_T(f) \lesssim \begin{cases} L^{\frac{1}{2\alpha}} \sqrt{T}, & \text{if } (\ell_t) \text{ convex}, \\ L^{\frac{1}{2\alpha}} \sqrt{T} \wedge L^{\frac{2}{2\alpha+1}} T^{\frac{1}{2\alpha+1}}, & \text{if } (\ell_t) \text{ exp-concave}. \end{cases}$$

Algorithm: Core Tree $\mathcal{T}_0 = (\mathcal{X}_n, (\mathcal{T}_{n,k})_{k \in [K]})_{n \in \mathcal{N}(\mathcal{T}_0)}$ associated to chaining tree predictors $(\hat{f}_{n,k})$, weights $(w_{n,k})$ and outputs average prediction at time $t \geq 1$

$$\hat{f}_t(x_t) = \sum_{n,k} w_{n,k,t} \hat{f}_{n,k,t}(x_t).$$

Local approximation: for $n \in \mathcal{N}(\mathcal{T}_0)$, each chaining tree predictor $\hat{f}_{n,k}, k \in [K]$ performs a local approximation over $\mathcal{X}_n \subset \mathcal{X}$ and is rooted at the k-th uniform grid point of $[-B, B]$, where the grid size is $K = \lceil 2B\sqrt{T} \rceil$, i.e. with precision $\varepsilon = T^{-\frac{1}{2}}$.



Algorithm 2: Locally Adaptive Online Regression at time t

Input : Chaining tree predictors $(\hat{f}_{n,k,t})$ and weights $(w_{n,k,t})$ of $\mathcal{T}_0$ at time t

Receive x_t and predict with predictors in $\mathtt{path}_{\mathcal{T}_0}(x_t)$, $\hat{f}_t(x_t) = \sum_{n,k} w_{n,k,t} \hat{f}_{n,k,t}(x_t)$;

Reveal gradient $(\partial_{w_{n,k,t}} \ell_t(\sum_{n,k} w_{n,k,t} \hat{f}_{n,k,t}(x_t)))$ and update weights $(w_{n,k,t})$ of $\mathcal{T}_0$;

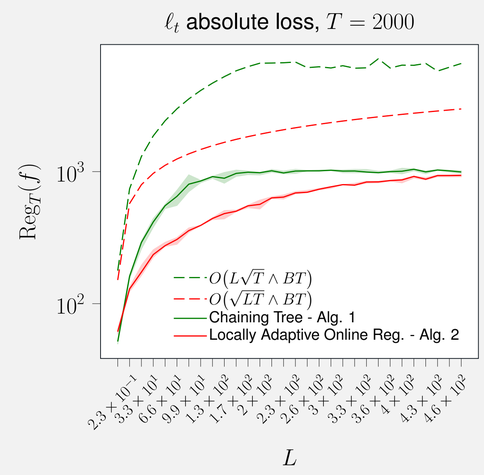
for $(n, k) \in path_{\mathcal{T}_0}(x_t) \times [K]$ **do**

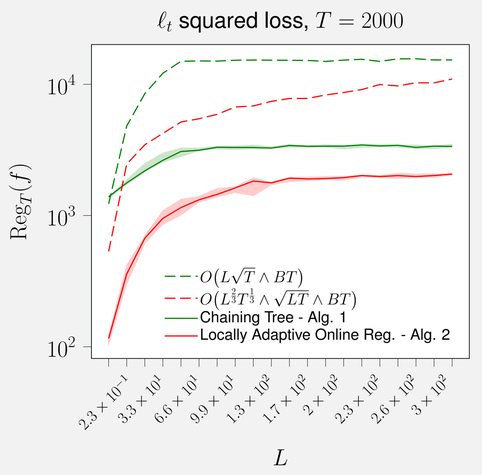
Reveal gradient $g_{n,k,t} = \ell_t'(\hat{f}_{n,k,t}(x_t))$;

Update and clip chaining tree predictor $\hat{f}_{n,k,t}$ using Algorithm 1 with $g_{n,k,t}$.

Output: Weights $(w_{n,k,t+1})$ and predictors $(\hat{f}_{n,k,t+1})$

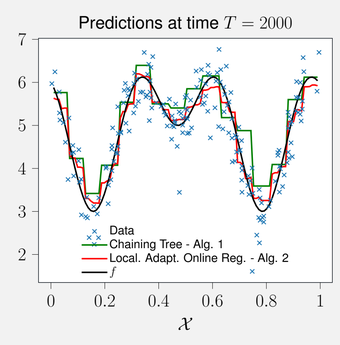
XP: $f(x) = \sin(10Lx) + \cos(5Lx) + 5, L \in [2^{-6}, 2^{-5}], x_t \sim \mathcal{U}([0, 1])$.

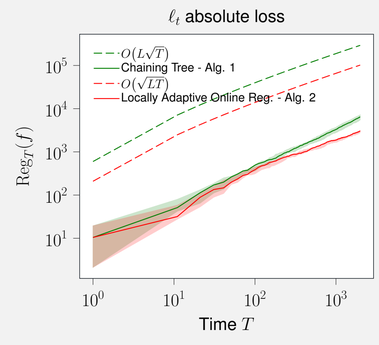


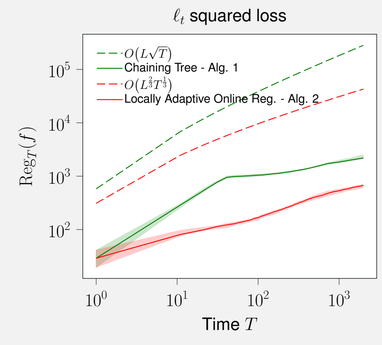


Numerical Experiments

Synthetic regression setting: data follows $f(x) + \varepsilon$, with $\varepsilon \sim 0.5 \cdot \mathcal{N}(0, 1)$, $f(x) = \sin(10x) + \cos(5x) + 5$, for $x \sim \mathcal{U}([0, 1])$ and $L = \sup_x |f'(x)| \leq 15$.







References

[1] Pierre Gaillard and Sébastien Gerchinovitz. "A chaining algorithm for online nonparametric regression". In: *Conference on Learning Theory*. PMLR. 2015, pp. 764–796.

[2] Ilja Kuzborskij and Nicolo Cesa-Bianchi. "Locally-adaptive nonparametric online learning". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 1679–1689.

[3] Francesco Orabona and Dávid Pál. "Coin betting and parameter-free online learning". In: *Advances in Neural Information Processing Systems* 29 (2016).

[4] Alexander Rakhlin and Karthik Sridharan. "Online non-parametric regression". In: *Conference on Learning Theory*. PMLR. 2014, pp. 1232–1264.