A chaining algorithm for online nonparametric regression

Pierre Gaillard\textsuperscript{1,2} and Sébastien Gerchinovitz\textsuperscript{3}

\textsuperscript{1}EDF R&D, Clamart, France
\textsuperscript{2}GREGHEC: HEC Paris - CNRS, Jouy-en-Josas, France
\textsuperscript{3}Institut de Mathématiques de Toulouse, Université Paul Sabatier, France
Setting: online regression with individual sequences

Online prediction protocol: at each round $t \in \mathbb{N}^*$,

1. The environment reveals the input $x_t \in \mathcal{X}$.
2. The forecaster chooses a prediction $\hat{y}_t \in \mathbb{R}$.
3. The environment reveals the observation $y_t \in \mathbb{R}$ and the forecaster incurs the square loss $(y_t - \hat{y}_t)^2$.

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

$$
\text{Reg}_T(\mathcal{F}) \triangleq \sum_{t=1}^{T} (y_t - \hat{y}_t)^2 - \inf_{f \in \mathcal{F}} \sum_{t=1}^{T} (y_t - f(x_t))^2.
$$

Individual sequences: our regret guarantees hold uniformly over all sequences $(x_t, y_t)_{t \geq 1}$ with bounded observations $y_t$. 
Contribution 1: chaining algorithm with Dudley bound

We design an explicit algorithm with Dudley-type regret bound (same vein as in Rakhlin and Sridharan 2014 but in a constructive fashion):

\[
\text{Reg}_T(\mathcal{F}) \leq \Box B^2 \left(1 + \log \mathcal{N}_\infty(\mathcal{F}, \gamma)\right) + \Box B \sqrt{T} \int_0^\gamma \sqrt{\log \mathcal{N}_\infty(\mathcal{F}, \varepsilon)} \, d\varepsilon
\]

where \( \log \mathcal{N}_\infty(\mathcal{F}, \varepsilon) \) is the metric entropy of \( \mathcal{F} \) in the sup norm.

**Corollaries:**

<table>
<thead>
<tr>
<th>Function class</th>
<th>Regret bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lipschitz on ([0, 1])</td>
<td>(T^{1/3})</td>
</tr>
<tr>
<td>(\beta)-Hölder on ([0, 1]), (\beta &gt; 1/2)</td>
<td>(T^{1/(2\beta+1)})</td>
</tr>
<tr>
<td>Sparse linear regression</td>
<td>(s \log(1 + dT/s))</td>
</tr>
</tbody>
</table>
Main intuitions behind our chaining algorithm

**Chaining technique:** 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 $k \geq 0$,

$$\sup_{f} \|\pi_k(f) - f\|_{\infty} \leq \gamma/2^k$$

and

$$\text{Card } \mathcal{F}^{(k)} = N_{\infty}(\mathcal{F}, \gamma/2^k)$$

$$\inf_{f \in \mathcal{F}} \sum_{t=1}^{T} (y_t - f(x_t))^2 = \inf_{f \in \mathcal{F}} \sum_{t=1}^{T} \left( y_t - \pi_0(f)(x_t) - \sum_{k=1}^{\infty} \left[ \pi_k(f) - \pi_{k-1}(f) \right](x_t) \right)^2$$

Our algorithm relies on a two-scale aggregation:

- **high-scale aggregation:** we use an Exponentially Weighted Average forecaster to mimic the best rough approximation $\pi_0(f) \in \mathcal{F}^{(0)}$;

- **low-scale aggregation:** new multi-scale version of Exponentiated Gradient to be competitive against all increments $\pi_k(f) - \pi_{k-1}(f)$.

**Previous works:** Cesa-Bianchi and Lugosi (1999, 2001) used chaining for absolute or log loss. For square loss, multi-scale linearization proves crucial.
Contribution 2: efficient algorithm for Hölder classes

**Computational issue:**
Without proper care, our algorithm is computationally intractable for Hölder classes (needs to update $\exp(O(T))$ weights at every round $t$).

**Solution:** we use computationally manageable coverings $\mathcal{F}^{(k)}$, $k \geq 0$:
- approximate any Lipschitz/Hölder function $f \in [0, 1] \rightarrow [-B, B]$ with piecewise constant/polynomial functions (level $k = 0$);
- refine the approximation via a dyadic discretization (levels $k \geq 1$).

<table>
<thead>
<tr>
<th>Function class</th>
<th>Regret bound</th>
<th>Time complexity</th>
<th>Space complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lipschitz on $[0, 1]$</td>
<td>$T^{1/3} \log T$</td>
<td>$T^{4/3} \log T$</td>
<td>$T^{4/3} \log T$</td>
</tr>
<tr>
<td>$\beta$-Hölder on $[0, 1]$</td>
<td>$T^{1/(2\beta+1)}(\log T)^{3/2}$</td>
<td>$\text{poly}(T)$</td>
<td>$\text{poly}(T)$</td>
</tr>
</tbody>
</table>
References

