Adaptive black-box optimization got easier: HCT only needs local smoothness

Xuedong Shang, Emilie Kaufmann, Michal Valko
xuedong.shang@inria.fr, emilie.kaufmann@univ-lille1.fr, michal.valko@inria.fr

Setting

- **Objective**: Find **a** maximum of an unknown function $f : \mathcal{X} \to \mathbb{R}$ with noisy observations.
- At each round t, a learner
  - evaluates a point $x_t \in \mathcal{X}$ and observes $r_t \triangleq f(x_t) + \varepsilon_t$,
  - recommends a point $x(t)$.
- **Performance measure**: *simple regret* $\triangleq f^\star - f(x(n))$.

Measure of complexity

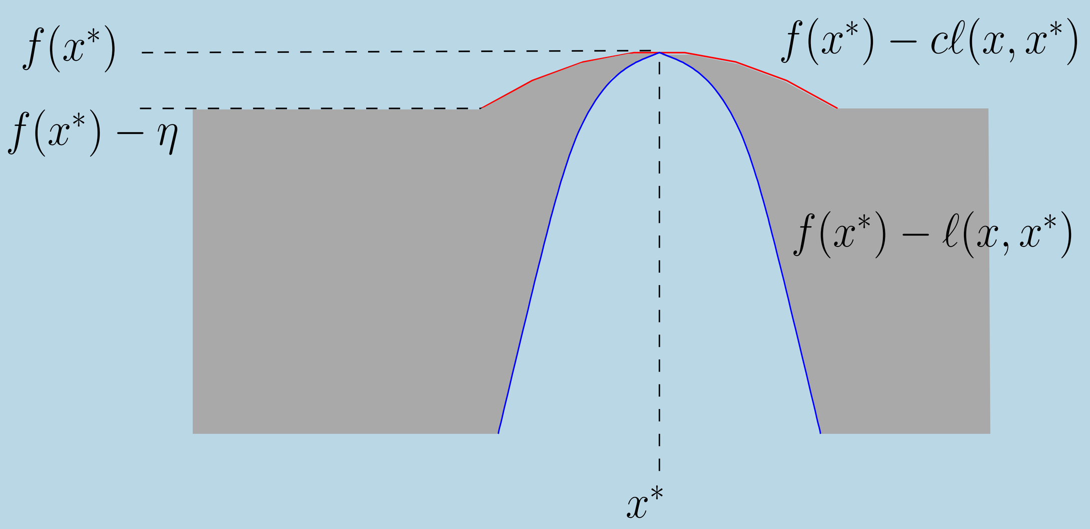
Definition 1 (near-optimality dimension w.r.t. $\mathcal{P}$)

$$d(\nu, \rho) \triangleq \inf\{d' \in \mathbb{R}^+ : \exists C > 0, \forall h \geq 0, \mathcal{N}_h(3\nu\rho^h) \leq C\rho^{-d'h}\}.$$

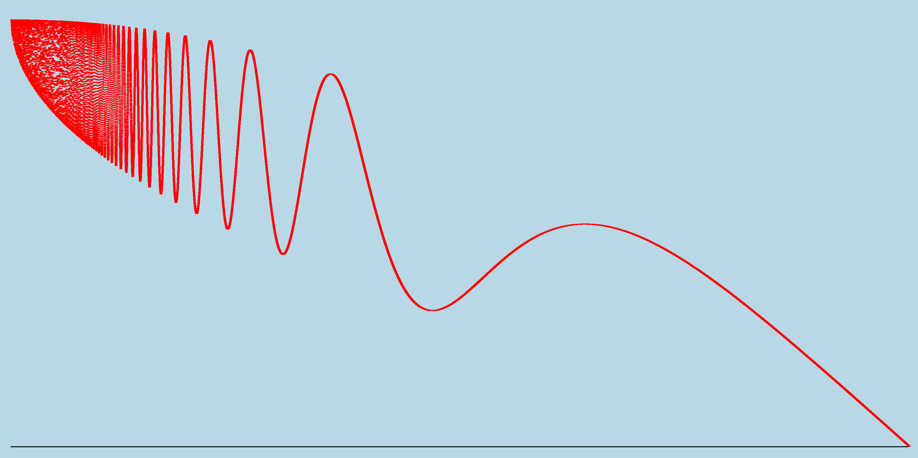
where $\mathcal{N}_h(3\nu\rho^h)$ is the number of cells $\mathcal{P}_{h,i}$ s.t. $\sup_{x \in \mathcal{P}_{h,i}} f(x) \geq f^\star - 3\nu\rho^h$.

Interpretation: $d(\nu, \rho)$ controls the amount of near-optimal cells → measures **how much information** $\mathcal{P}$ gives us about f.

→ **Examples of functions with different d values**



same upper and lower envelope
$\boldsymbol{d = 0}$

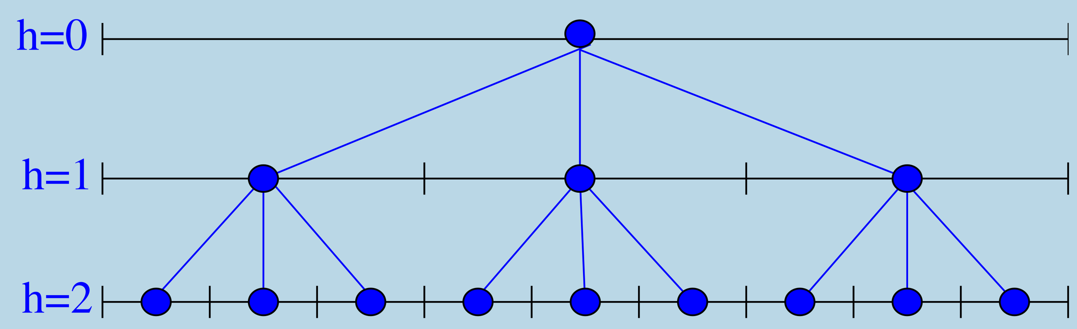


$f(x) = 1 - \sqrt{x} + (-x^2 + \sqrt{x}) \cdot (\sin(1/x^2) + 1)/2$
$\boldsymbol{d = \frac{1}{2}}$

Standard partitioning

Hierarchical bandits rely on a standard **hierarchical partitioning** $\mathcal{P} = \{\mathcal{P}_{h,i}\}$ defined recursively as

$$\mathcal{P}_{0,1} \triangleq \mathcal{X}, \mathcal{P}_{h,i} \triangleq \cup_{j=0}^{K-1} \mathcal{P}_{h+1,Ki-j}.$$



Assumptions

Assumption 1 *Let $x^\star$ be a global maximizer and $i_h^\star$ be the index of the only cell at depth h that contains $x^\star$. There exist $\nu > 0$, $\rho \in (0,1)$ s.t.*

$$\forall h \geq 0, \forall x \in \mathcal{P}_{h,i_h^\star}, \quad f(x) \geq f^\star - \nu\rho^h.$$

It is a one-side local Lipschitz-type of assumption that naturally covers large class of functions. It is constraining f only along the optimal path and **does not rely on any metric!**

→ Previous algorithms that **depend on a metric**:

	global	local
known	Zooming, HOO	DOO, HCT
unknown	TaxonomyZoom	SOO, StoSOO, ATB

→ POO(HCT) and POO(HOO):
unknown local smoothness **without metric!**

Contributions

Context: Non-trivial to provide a sublinear regret bound for HOO under Assumption 1.
→ We propose POO on top of HCT with an analysis under Assumption 1.

How and Why

How it works?

- HCT traverses an *optimistic path* P_t by repeatedly selecting cells that have a larger U-value until a leaf or a node that is sampled less than $\tau_h(t)$ times.
- POO launches several instances of HCT in parallel with different smoothness and selects the instance with the best performance.

Why it works?

- HOO could induce a very deep covering tree, while producing too many neither near-optimal nor sub-optimal nodes.
- HCT, while having a limited depth, has the possibility to control the number of such nodes.
- Few HCT instances are needed - $\mathcal{O}(\log n)$.

References

[1] POO: Jean-Bastien Grill, Michal Valko, and Rémi Munos. *Black-box optimization of noisy functions with unknown smoothness.* In Neural Information Processing Systems, 2015.

[2] HCT: Mohammad Gheshlaghi Azar, Alessandro Lazaric, and Emma Brunskill. *Online stochastic optimization under correlated bandit feedback.* In International Conference on Machine Learning, 2014.

[3] HOO: Sébastien Bubeck, Rémi Munos, Gilles Stoltz, and Csaba Szepesvári. *X-armed bandits.* Journal of Machine Learning Research, 12:1587-1627, 2011.

Algorithms

HCT

```
Parameters: ν ρ, c, P, δ
Initialization:
  T_1 ← {(0,1), (1,1), (1,2)}
  H(1) ← 1, U_{1,1}(1) ← U_{1,2}(1) ← +∞
for t = 1 ··· n do
  if t = 2^{⌈log(t)⌉} then
    Update the whole covering tree T_t
  end if
  (h_t, i_t), P_t ← OptTraverse(T_t)
  Evaluate x_{h_t,i_t} and obtain r_t
  Update μ̂_{h_t,i_t}(t) and U_{h_t,i_t}(t)
  UpdateB(T_t, P_t, (h_t, i_t))
  Compute τ_{h_t}(t)
  if T_{h_t,i_t}(t) ≥ τ_{h_t}(t) and (h_t, i_t) is a leaf then
    Expand((h_t, i_t))
  end if
end for
```

POO(HCT)

```
Parameters: K, P, ρ_max, ν_max
Initialization:
  D_max ← ln K / ln(1/ρ_max)
  n ← 0, N ← 1, S ← {(ν_max, ρ_max)}
while budget still available do
  while N ≥ (1/2) D_max ln(n/(ln n)) do
    for i ← 1, ..., N do
      s ← (ν_max, ρ_max^{2N/(2i+1)})
      Start HCT(s) run for n/N times
    end for
    n ← 2n, N ← 2N
  end while
  Run each HCT(s) once
  n ← n + N
end while
s* ← argmax_{s∈S} μ̂[s]
Output: A point sampled u.a.r. from the points evaluated by HCT(s*)
```

Analysis

Theorem 1 *Assume that function f satisfies Assumption 1. Then, using the recommendation strategy $x(n) \sim \mathcal{U}(\{x_1, \ldots, x_n\})$, the simple regret of* HCT *after n rounds is bounded as*

$$\mathbb{E}[S_n^{\texttt{HCT}}] \leq \mathcal{O}\left((\log n)^{1/(d+2)} n^{-1/(d+2)}\right).$$

The previous result can then be plugged into POO's analysis, helping us getting the following bound.

Theorem 2 *The simple regret of* POO(HCT) *is bounded as*

$$\mathbb{E}[S_n^{\texttt{POO}}] \leq \mathcal{O}\left((\log^2 n)/n)^{1/(d(\nu^\star,\rho^\star)+2)}\right),$$

where $(\nu^\star, \rho^\star)$ is the couple of parameters corresponding to the best performing HCT instance.