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 \( f : \mathcal{X} \rightarrow \mathbb{R} \) with noisy observations.
- At each round \( t \), a learner evaluates a point \( x_t \in \mathcal{X} \) and observes \( r_t \triangleq f(x_t) + \varepsilon_t \).
- Performance measure: simple regret \( \triangleq f^* - f(x(\tau)) \).

Measure of complexity

Definition 1 (near-optimality dimension w.r.t. \( P \))

\[
d(\nu, \rho) \triangleq \inf \{ \delta \in \mathbb{R}^+ : \exists C > 0, \forall h \geq 0, N_h(3\nu \rho^h) \leq C \rho^{-d h} \}.
\]

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

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

→ Examples of functions with different \( d \) values

\[
f(x) = 1 - \sqrt{x + (x^2 + \sqrt{x}) \cdot (\sin(1/x^2) + 1)/2}
\]

\[
d = 0
\]

\[
d = \frac{1}{2}
\]

same upper and lower envelope

Contributions

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

How and Why

How it works?
- HCT traverses an optimistic path \( P_1 \) by repeatedly selecting cells that have a larger \( U \)-value until a leaf or a node that is sampled less than \( \tau_n(t) \) times.
- PDD 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 - \( O(\log n) \).

Algorithms

HCT

Parameters: \( \nu, \rho, \rho, c, \mathcal{P}, \delta \)

Initialization:

\[
T_1 \leftarrow \{(0,1), (1,1), (1,2)\}
\]

\[
H(1) \leftarrow 1, U_{1,1}(1) \leftarrow U_{1,2}(1) \leftarrow +\infty
\]

for \( t = 1 : \cdots \cdot n \) do

if \( t \geq 2 \log(\varepsilon) \) then

Update the whole covering tree \( T_{\tau} \)

end if

\( (h, i), P_t \leftarrow \text{OptTraverse}(T_{\tau}) \)

Evaluate \( x_{h,i}, s \) and obtain \( r_t \)

Update \( \rho_{h,i}(t) \) and \( U_{h,i}(t) \)

Update \( \mathcal{E}(T_{\tau}, P_{t}, (h, i)) \)

Compute \( \tau_n(t) \)

if \( T_{h,i}(t) \geq \tau_n(t) \) and \( (h, i) \) is a leaf then

Expand \((h, i)\)

end if

end for

PDD(HCT)

Parameters: \( K, \mathcal{P}, \rho_{\max}, \nu_{\max} \)

Initialization:

\[
D_{\max} \leftarrow \ln K / \ln(1/\rho_{\max})
\]

\( n \leftarrow 0, N \leftarrow 1, S \leftarrow \{(\nu_{\max}, \rho_{\max})\} \)

while budget still available do

while \( N > \frac{1}{\sqrt{D_{\max}}} \ln(n/\ln n) \) do

for \( i = 1, \ldots, N \) do

\( s \leftarrow \{\nu_{\max}, \rho_{\max} 2N/(2i+1)\} \)

Start HCT\( (s) \) run for \( \frac{\nu}{\rho} \) times

end for

end while

Run each HCT\( (s) \) once

\( n \leftarrow n + N \)

end while

\( x^* \leftarrow \arg\max_{s \in S} \hat{\mu}(s) \)

Output: A point sampled u.a.r. from the points evaluated by HCT\( (x^*) \)

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

\[
\mathbb{E}[S_{\text{HCT}}^n] \leq O\left( (\log n)^{1/(d+2)} n^{1/(d+2)} \right).
\]

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

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

\[
\mathbb{E}[S_{\text{PDD(HCT)}}^n] \leq O\left( (\log^2 n) n^{1/(d(\nu^*, \rho^*))+2} \right),
\]

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

References

