Optimum-statistical Collaboration Towards Efficient Black-box Optimization

(Global Optimization-Part II)

Wenjie Li* Chi-Hua Wang*, Guang Cheng

li3549@purdue.edu

Department of Statistics
Purdue University

November 19, 2021

Introduction

Suppose we want to optimize an unknown function $f(x)$, on a parameter domain $\mathcal{X}$, without the gradient of the function. What should we do?

Introduction (cont.)

For simplicity now, let's say the domain is $\mathcal{X} = [0, 1]$, and the function is one dimensional.

We can do the Bisection algorithm!

Pull once from [0, 0.5], lets say at $x = 0.25$, get value v_1

Pull once from [0.5, 1], lets say at $x = 0.75$, get value v_2

Pull once from [0, 0.25], get value v_3

Pull once from [0, 0.75], get value v_4

...

But we may need some smoothness assumptions, so that the pulled points can represent the sub-domains. We will talk about this later.

Introduction (cont.)

Pull once from [0, 0.5], lets say at $x = 0.25$, get value v_1

Pull once from [0.5, 1], lets say at $x = 0.75$, get value v_2

Pull once from [0, 0.25], get value v_3

Pull once from [0.25, 0.5], get value v_4

...

What does the above process generate?

A tree!

Introduction (cont.)

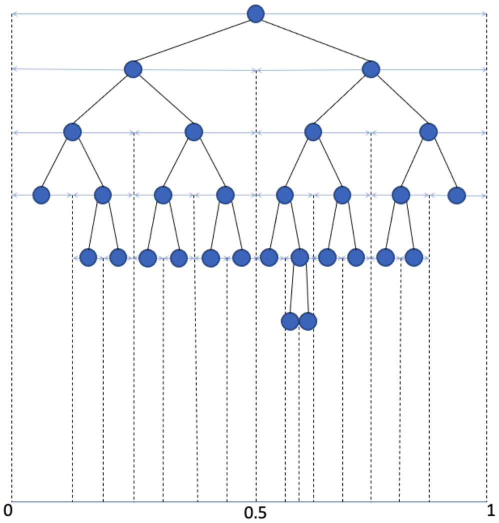
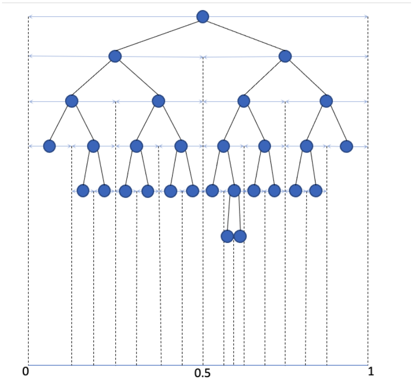


Figure 1: Generated Tree

Introduction (cont.)

Can we do something like $[0, 0.3], [0.3, 1]$ on each level?



Notations

In GO, we optimize an unknown and costly-to-evaluate function $f : \mathcal{X} \to \mathbb{R}$ based on n noisy evaluations, that can be sequentially selected.

Example: Hyper-parameter tuning, such as the initial step size for SGD.

Notations (cont.)

Let $\mathcal{X}$ be a measurable space. At each round t, a learner selects a point $x_t \in \mathcal{X}$ and observes a reward $r_t \triangleq f(x_t) + \varepsilon_t$, bounded by [0,1] from the environment where the noise ε_t is assumed to be independent from previous observations and such that $\mathbb{E}[\varepsilon_t \mid x_t] = 0$.

Notations (cont.)

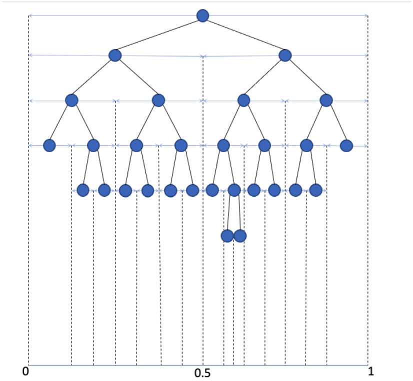
After n evaluations, the algorithm outputs a guess for the maximizer, denoted by $x(n)$. We assume that there exists at least one $x^* \in \mathcal{X}$ s.t. $f(x^*) \triangleq \sup_{x\in\mathcal{X}} f(x)$ denoted by f^* in the following. We measure the performance by the cumulative regret,

$$R_n \triangleq nf^* - \sum_{t=1}^{n} f(x_t)$$

Partition

The algorithms rely on the existence of hierarchical partitioning $\mathcal{P} \triangleq \{\mathcal{P}_{h,i}\}_{h,i}$ defined recursively, where

$$\mathcal{P}_{0,1} = \mathcal{X}, \quad \mathcal{P}_{h,i} = \bigcup_{j=0}^{1} \mathcal{P}_{h+1,2i-j}$$



Assumptions

What other people (e.g. [1], [2]) have assumed...

Assumption

(local smoothness w.r.t. $\mathcal{P}$) For $x^\star$ be a global maximizer, we denote by $i_h^\star$ be the index of the only cell at depth h that contains $x^\star$. Then, there exist a global maximizer $x^\star$ and two constants $\nu > 0, \rho \in (0, 1)$ s.t.,

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

Assumptions (cont.)

Assumption

(Local smoothness w.r.t. $\mathcal{P}$) For $x^\star$ be a global maximizer, we denote by $i_h^\star$ be the index of the only cell at depth h that contains $x^\star$. Then, there exist a global maximizer $x^\star$ and two constants $\nu > 0, \rho \in (0, 1)$ s.t.,

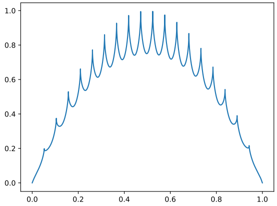
$$\forall h \geq 0, \forall x \in \mathcal{P}_{h,i_h^*}, \quad f(x) \geq f^\star - \nu\rho^h$$

Consider the standard binary partition on $\mathcal{X} = [0, 1]$.

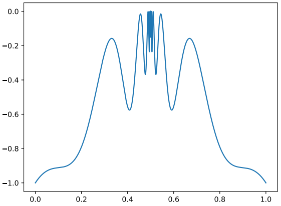
The above assumption (roughly) means that when we get $(0.5)^h$ close to (one) maximizer, the function value is $\nu\rho^h$ close to the optimum.

Assumptions (cont.)

A lot of functions satisfy this assumption, some one dimensional examples are



(a) The Garland Function



(b) The Double Sine Function

And any function that has can be written as a combination of $f_\alpha(x) = \|x - x^*\|^\alpha$.

Assumptions (cont.)

However, there are functions and partitions that do no satisfy the assumption, one example is $g(x) = 1 + 1/\ln x$ on $(0, 1/e)$ with the standard partition

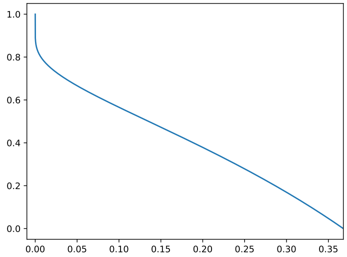


Figure 3: The Counter Example Function

Why? Because the function decreases too fast around zero

Assumptions (cont.)

Recall the assumption again,

Assumption

(local smoothness w.r.t. $\mathcal{P}$) For $x^\star$ be a global maximizer, we denote by $i_h^\star$ be the index of the only cell at depth h that contains $x^\star$. Then, there exist a global maximizer $x^\star$ and two constants $\nu > 0, \rho \in (0,1)$ s.t.,

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

If we use the standard partitioning, since $g(x)$ is monotone, the minimum of the function in the optimal node is at the boundary point, i.e., $x_0 = (1/2)^h * 1/e$

Assumptions (cont.)

Substitute the above choice of x_0 back to the function, we get

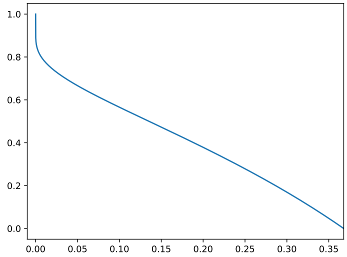
$$g(x_0) = 1 + \frac{1}{\ln x} = 1 - \frac{1}{1 + h \ln 2} \leq 1 - \frac{1}{2h}$$

Note that the optimum is $g^* = 1$. Can we find some ν, ρ so that $\nu \rho^h \geq \frac{2}{h}$ always holds for any $h \geq 0$? No.

In other words, the assumption people use limit the amount of change in the function, i.e., it cannot change too fast around the optimum.

Assumptions (cont.)

Is this function hard to optimize?



There might exists a good partition, so that $g(x)$ and the partition satisfies the above assumption. However, since $g(x)$ is unknown before the optimization, how do we choose the good partition?

The General Framework

Assumption

(General local smoothness w.r.t. $\mathcal{P}$) For $x^\star$ be a global maximizer, we denote by $i_h^\star$ be the index of the only cell at depth h that contains $x^\star$. Then, there exist a global maximizer $x^\star$ and a function $\xi(h)$ s.t.,

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

The General Framework (cont.)

Another notion that people use is the near-optimality dimension...

Definition

(Near-optimality dimension w.r.t. $\mathcal{P}$) For any $\nu > 0$, $C > 1$, and $\rho \in (0,1)$, we define the near-optimality dimension of f with respect to $\mathcal{P}$ as

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

where $\mathcal{N}_h(\varepsilon)$ is the number of cells $\mathcal{P}_{h,i}$ such that $\sup_{x \in \mathcal{P}_{h,i}} f(x) \geq f^\star - \varepsilon$.

The above is too counter-intuitive, and hard to understand. Let me first introduce the general notion and then explain why.

The General Framework (cont.)

Definition

(Near-optimality dimension w.r.t. $\mathcal{P}$) For any $a > 0$, $C > 1$, and function $\xi(h) \in (0, 1]$ for $h \geq 1$, we define the near-optimality dimension of f with respect to $\mathcal{P}$ as

$$d(a, C, \xi(h)) \triangleq \inf \left\{ d' \in \mathbb{R}^+ : \forall h \geq 0, \mathcal{N}_h \left(a\xi(h) \right) \leq C\xi(h)^{-d'} \right\},$$

where $\mathcal{N}_h(\varepsilon)$ is the number of cells $\mathcal{P}_{h,i}$ such that $\sup_{x \in \mathcal{P}_{h,i}} f(x) \geq f^\star - \varepsilon$.

What does the above notion imply?

It implies that there exists a d, such that the number of near-optimal nodes (($a\xi(h)$) close to optimum), is bounded by $C\xi(h)^{-d}$

The General Framework (cont.)

It implies that there exists a d, such that the number of near-optimal nodes $((a\xi(h))$ close to optimum), is bounded by $C\xi(h)^{-d}$

Examples. Assume that d exists in each case

1. $\xi(h) = 1/2$, then $\mathcal{N}_h(a\xi(h)) \leq 2^d$ (Constant)
2. $\xi(h) = 1/h$, then $\mathcal{N}_h(a\xi(h)) \leq Ch^d$. (Polynomially increasing)
3. $\xi(h) = \rho^h$, then $\mathcal{N}_h(a\xi(h)) \leq C\rho^{-hd}$. (Exponentially increasing)

The General Framework (cont.)

Remember the function we use for general local smoothness is also $\xi(h)$, this is not a coincidence.

When we are $a\xi(h)$-close to the optimum, we will see $N_h(a\xi(h))$ number of near-optimal regions, and we want this number to be upper bounded by some function.

1. $\xi(h) = 1/2$, then $\mathcal{N}_h(a\xi(h)) \leq 2^d$ (Constant)
2. $\xi(h) = 1/h$, then $\mathcal{N}_h(a\xi(h)) \leq Ch^d$. (Polynomially increasing)
3. $\xi(h) = \rho^h$, then $\mathcal{N}_h(a\xi(h)) \leq C\rho^{-hd}$. (Exponentially increasing)

The difficulty of optimizing a function is determined by

1. Smoothness
2. Number of near-optimal regions

The General Framework (cont.)

However, previous analyses all rely on $\xi(h) = \rho^h$. It means that they cannot be applied, even to easy functions like $g(x) = 1 + 1/\ln x$.

Can we generalize?

The General Framework (cont.)

Definition

(Resolution Descriptor) Define $\mathtt{OE}_h$ to be the *resolution* for each layer h, which is a function that upper-bounds the change of f around one optimum according to the partition and measures the current optimization error, i.e.,

$$\forall h \geq 0, \forall x \in \mathcal{P}_{h,i_h^*}, f(x) \geq f(x^*) - \mathtt{OE}_h$$

where $\mathcal{P}_{h,i_h^*}$ is the node on layer h that contain the global maximizer x^*. If multiple maximizers exist, the above relation only needs to be satisfied by one.

The General Framework (cont.)

Definition

(Uncertainty Quantifier) Define $\mathtt{SE}_{h,i}(t)$ to be the *uncertainty estimate* for each node $\mathcal{P}_{h,i}$ at time t, which is a function that upper-bounds the current statistical error of the average $\widehat{\mu}_{h,i}(t)$ of the rewards obtained with high probability, i.e., the event

$$\mathcal{A}_t = \left\{ \forall (h,i), |\widehat{\mu}_{h,i}(t) - f(x_{h,i})| \leq \mathtt{SE}_{h,i}(t) \right\}$$

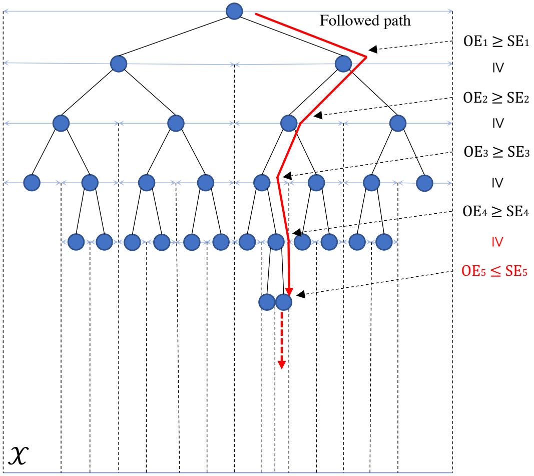
is a high probability event.

The General Framework (cont.)

The General Algorithm

- **Input:** Hierachical partition $\mathcal{P}$, resolution descriptor $\texttt{OE}_h$, Uncertainty quantifier $\texttt{SE}_{h,i}(t)$.
- **Step 1:** Refresh the confidence at some specific time steps to update all $\texttt{SE}_{h,i}(t)$ in the tree.
- **Step 2:** Find the optimal node $\mathcal{P}_{h_t,i_t}$ at time t that satisfies $\texttt{OE}_{h_t} \leq \texttt{SE}_{h_t,i_t}(t)$ and pull $\mathcal{P}_{h_t,i_t}$.
- **Step 3:** If $\texttt{OE}_{h_t} \geq \texttt{SE}_{h_t,i_t}(t)$ after the pull, expand $\mathcal{P}_{h_t,i_t}$ and explore deeper.

The General Framework (cont.)



$$\mathtt{OE}_1 \geq \mathtt{SE}_1 > \mathtt{OE}_2 \geq \mathtt{SE}_2 \geq \cdots \geq \mathtt{OE}_h \geq \mathtt{SE}_h \geq \cdots \tag{1}$$

The General Framework (cont.)

Theorem

(General Regret Bound) *Suppose that under a sequence of probability events* $\{\mathcal{E}_t\}_{t=1,2,\dots}$ *at each time* t*, the designed policy to select the optimal node* $\mathcal{P}_{h_t,i_t}$ *satisfies* $f^* - f(x_{h_t,i_t}) \leq a \cdot \max\{\mathtt{SE}_{h_t,i_t}(t),\ \mathtt{OE}_{h_t}\}$*, where* $a > 0$ *is an absolute constant. Let* $H(n)$ *denotes the tree depth at time* n*. Then for any* $\overline{H} \in [1, H(n))$ *we have the following bound on the expected regret*

$$\mathbb{E}[R_n] \leq \sqrt{2n\log(4n^3)} + \frac{1}{4n^2} + \sum_{t=1}^{n} \mathbb{P}(\mathcal{E}_t^c) + a \sum_{\overline{H}+1}^{H(n)} \sum_{i \in \mathcal{I}_h(n)} \int_1^{T_{h,i}(n)} \mathtt{SE}_{h,i}(s)ds$$

$$+ 2aC \sum_{h=1}^{\overline{H}} (\mathtt{OE}_{h-1})^{-\bar{d}} \int_1^{T_h(n)} \max_i \mathtt{SE}_{h,i}(s)ds$$

where and $\bar{d} := d(a, C, \mathtt{OE}_{h-1})$ *is the near-optimality dimension, and* $T_h(n) = \max_i T_{h,i}(n)$

Example

Examples of $\mathtt{OE}$.

1. $\mathtt{OE}_h = 1/2$
2. $\mathtt{OE}_h = 1/h$
3. $\mathtt{OE}_h = \rho^h$ (What people have used before)

Example (cont.)

Examples of $\texttt{SE}$.

Nonadaptive Quantifier ($\texttt{HCT}$) Azar et al. proposed in their High Confidence Tree ($\texttt{HCT}$) algorithm to use an uncertainty quantifier of the following form [3]

$$\texttt{SE}_{h,i}(t) \equiv bc\sqrt{\frac{\log\left(1/\tilde{\delta}(t^+)\right)}{T_{h,i}(t)}}$$

where t^+ is the time step to update the confidence, which is a variable related to t and c is a tuning constant.

Example (cont.)

Variance Adaptive Quantifier (VHCT) Based on our framework of the statistical collaboration, a tighter measure of the statistical uncertainty can boost the performance of the optimization algorithm, as the goal in Eqn. (1) can be reached faster. We therefore propose the following variance adaptive uncertainty quantifier, and naturally the VHCT algorithm in the next subsection, which is an adaptive variant of HCT.

$$\mathrm{SE}_{h,i}(t) \equiv c\sqrt{\frac{2\widehat{\mathbb{V}}_{h,i}(t)\log\left(1/\tilde{\delta}(t^{+})\right)}{T_{h,i}(t)}} + \frac{3bc^2\log\left(1/\tilde{\delta}(t^{+})\right)}{T_{h,i}(t)} \qquad (2)$$

Regret Analysis

Theorem

Assume that the objective function f satisfies the General Local Smoothness Assumption with $\phi(h) = \nu\rho^h$ for two constants $\nu > 0, \rho \in (0,1)$. The expected cumulative regret of Algorithm VHCT is upper bounded by

$$\mathbb{E}[R_n] \leq 2\sqrt{2n\log(4n^3)} + C_1 V_{\max}^{\frac{1}{d_1+2}} n^{\frac{d_1+1}{d_1+2}} (\log n)^{\frac{1}{d_1+2}} + C_2 n^{\frac{d_1}{d_1+2}} \log n.$$

where C_1 and C_2 are two constants and the near-optimality dimension $d_1 = d(3\nu, C, \rho^h)$.

Regret Analysis (cont.)

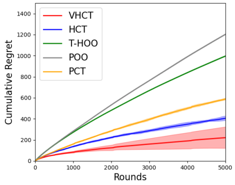
Theorem

Assume that the objective function f satisfies the General Local Smoothness Assumption with $\phi(h) = 1/h$. The expected cumulative regret of Algorithm VHCT is upper bounded by

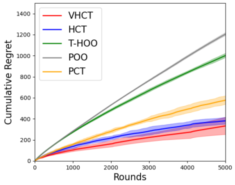
$$\mathbb{E}[R_n] \leq 2\sqrt{2n\log(4n^3)} + \bar{C}_1 V_{\max}^{\frac{1}{2d_2+3}} n^{\frac{2d_2+2}{2d_2+3}} (\log n)^{\frac{1}{2d_2+3}} \\ + \bar{C}_2 n^{\frac{2d_2+1}{2d_2+3}} \log n.$$

where $\bar{C}_1$ and $\bar{C}_2$ are two constants and the near-optimality dimension $d_2 = d(1, C, 1/h)$.

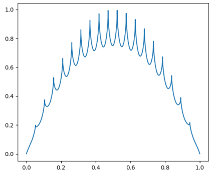
Experiments



(a) Low noise



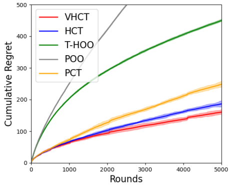
(b) Moderate noise



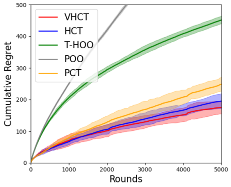
(c) Garland function

Figure 4: Cumulative regret of different algorithms on the Garland function $G(x)$

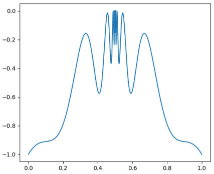
Experiments (cont.)



(a) Low noise



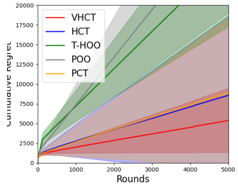
(b) Moderate noise



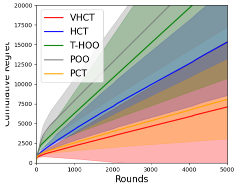
(c) Doublesine function

Figure 5: Cumulative regret of different algorithms on the Double-sine function $D(x)$

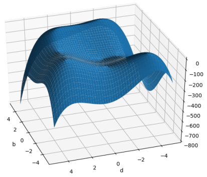
Experiments (cont.)



(a) Low noise



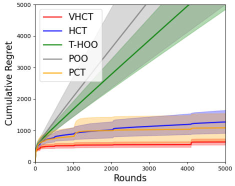
(b) Moderate noise



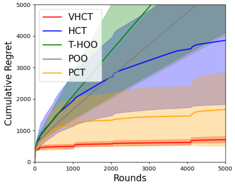
(c) Himmelblau function

Figure 6: Cumulative regret of different algorithms on the Himmelblau function

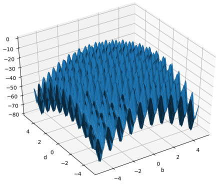
Experiments (cont.)



(a) Low noise



(b) Moderate noise



(c) Rastrigin function

Figure 7: Cumulative regret of different algorithms on the Rastrigin function

For more experimental results on different objective functions, please refer to our paper https://arxiv.org/abs/2106.09215

Open Questions

- Is there a uniform upper bound for different smoothness functions? (yes)
- Are the upper bounds we have derived tight? (almost)
- Is there an algorithm that works without prior knowledge of the smoothness? (Not sure yet)
- Robustness, stability, etc.
- How to apply these theoretical work to real life applications, e.g., NAS?

Bibliography

[1] J.-B. Grill, M. Valko, R. Munos, and R. Munos, “Black-box optimization of noisy functions with unknown smoothness,” in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2015.

[2] P. L. Bartlett, V. Gabillon, and M. Valko, “A simple parameter-free and adaptive approach to optimization under a minimal local smoothness assumption,” in *30th International Conference on Algorithmic Learning Theory*, 2019.

[3] M. G. Azar, A. Lazaric, and E. Brunskill, “Online stochastic optimization under correlated bandit feedback,” in *International Conference on Machine Learning*, PMLR, 2014, pp. 1557–1565.