Packing-based algorithms for instance-dependent error bounds in optimization and level set estimation

François Bachoc

Institut de Mathématiques de Toulouse
Université Paul Sabatier
France

Joint work with Sébastien Gerchinovitz and Tommaso Cesari (ANITI and IRT, Toulouse)

Toulouse-Berne
May 2022

Why this talk ?

Sequential procedures for costly black-box functions

- Gaussian processes : widely used sequential procedures (Efficient Global Optimization (EGO), Stepwise Uncertainty Reduction (SUR),...) [Jones et al., 1998, Chevalier et al., 2014].
- Theory : open questions remain despite existing work [Vazquez and Bect, 2010, Bull, 2011, Srinivas et al., 2012, Bect et al., 2019].
- "Machine learning" : theoretical results tend to be more refined, exponential rates, instance-dependent rates,... [Munos, 2011, de Freitas et al., 2012].
- But are these latter algorithms as "practical" for costly black-box functions ?

This talk

Presentation of two contributions in "machine learning".

F. Bachoc, T. Cesari and S. Gerchinovitz, “The sample complexity of level set approximation” AISTATS 2021 - oral presentation

F. Bachoc, T. Cesari and S. Gerchinovitz, “Instance-dependent bounds for zeroth-order Lipschitz optimization with error certificates” NeurIPS 2021

$\Longrightarrow$ Can be bridged with Gaussian processes ?

Why this talk ?

Sequential procedures for costly black-box functions

- Gaussian processes : widely used sequential procedures (Efficient Global Optimization (EGO), Stepwise Uncertainty Reduction (SUR),...) [Jones et al., 1998, Chevalier et al., 2014].
- Theory : open questions remain despite existing work [Vazquez and Bect, 2010, Bull, 2011, Srinivas et al., 2012, Bect et al., 2019].
- "Machine learning" : theoretical results tend to be more refined, exponential rates, instance-dependent rates,... [Munos, 2011, de Freitas et al., 2012].
- But are these latter algorithms as "practical" for costly black-box functions ?

This talk

Presentation of two contributions in "machine learning".

F. Bachoc, T. Cesari and S. Gerchinovitz, “The sample complexity of level set approximation” AISTATS 2021 - oral presentation

F. Bachoc, T. Cesari and S. Gerchinovitz, “Instance-dependent bounds for zeroth-order Lipschitz optimization with error certificates” NeurIPS 2021

$\Longrightarrow$ Can be bridged with Gaussian processes ?

1 Non-certified level set estimation

2 Certified optimization

Problem, motivations, related work

Problem

Approximating $\{\boldsymbol{x} : f(\boldsymbol{x}) = a\} \subset [0,1]^d$.

- $f : [0,1]^d \to \mathbb{R}$ unknown in some known smoothness class.
- $a \in \mathbb{R}$ a fixed known threshold.

Motivation

Determining parameters that result in a given outcome (computer experiments, uncertainty quantification, nuclear engineering, coastal flooding, etc).

Related work

- Gaussian process models : [Chevalier et al., 2014], [Azzimonti et al., 2021], [Gotovos et al., 2013].
- Global optimization algorithms : [Munos, 2011], [Bubeck et al., 2011].

Problem

Approximating $\{\boldsymbol{x} : f(\boldsymbol{x}) = a\} \subset [0,1]^d$.

- $f : [0,1]^d \to \mathbb{R}$ unknown in some known smoothness class.
- $a \in \mathbb{R}$ a fixed known threshold.

Motivation

Determining parameters that result in a given outcome (computer experiments, uncertainty quantification, nuclear engineering, coastal flooding, etc).

Related work

- Gaussian process models : [Chevalier et al., 2014], [Azzimonti et al., 2021], [Gotovos et al., 2013].
- Global optimization algorithms : [Munos, 2011], [Bubeck et al., 2011].

Problem, motivations, related work

Problem

Approximating $\{\boldsymbol{x} : f(\boldsymbol{x}) = a\} \subset [0,1]^d$.

- $f : [0,1]^d \to \mathbb{R}$ unknown in some known smoothness class.
- $a \in \mathbb{R}$ a fixed known threshold.

Motivation

Determining parameters that result in a given outcome (computer experiments, uncertainty quantification, nuclear engineering, coastal flooding, etc).

Related work

- Gaussian process models : [Chevalier et al., 2014], [Azzimonti et al., 2021], [Gotovos et al., 2013].
- Global optimization algorithms : [Munos, 2011], [Bubeck et al., 2011].

Protocol and objective

Online protocol

For $n = 1, 2, \ldots$:

1. pick the next query point $\boldsymbol{x}_n$,
2. observe the value $f(\boldsymbol{x}_n)$,
3. output an approximating set S_n.

Our goal

Quantifying the sample complexity, i.e., smallest number of evaluations of f needed to

$$\{\boldsymbol{x} : f(\boldsymbol{x}) = a\} \subset S_n \subset \{\boldsymbol{x} : |f(\boldsymbol{x}) - a| \leq \varepsilon\}.$$

For some error level $\varepsilon > 0$.

Protocol and objective

Online protocol

For $n = 1, 2, \ldots$:

1. pick the next query point $\boldsymbol{x}_n$,
2. observe the value $f(\boldsymbol{x}_n)$,
3. output an approximating set S_n.

Our goal

Quantifying the sample complexity, i.e., smallest number of evaluations of f needed to

$$\{\boldsymbol{x} : f(\boldsymbol{x}) = a\} \subset S_n \subset \left\{\boldsymbol{x} : |f(\boldsymbol{x}) - a| \leq \varepsilon\right\}.$$

For some error level $\varepsilon > 0$.

A hard problem

Definition

The packing number of a non-empty set E is

$$\mathcal{N}(E,\varepsilon) := \sup\left\{k \in \mathbb{N} : \exists\, \boldsymbol{x}_1,\ldots,\boldsymbol{x}_k \in E,\ \min_{i\neq j} \left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|_\infty > \varepsilon\right\}.$$

Theorem

If f is a non-constant continuous function, for any $\min(f) < a < \max(f)$,

$$\mathcal{N}\Big(\{f = a\},\varepsilon\Big) \gtrsim \frac{1}{\varepsilon^{d-1}}$$

as $\varepsilon \to 0$.

Not surprising, the level set is defined by a single equation in d unknowns.

A hard problem

Definition

The packing number of a non-empty set E is

$$\mathcal{N}(E,\varepsilon) := \sup\Big\{k \in \mathbb{N} : \exists\, \boldsymbol{x}_1,\dots,\boldsymbol{x}_k \in E,\ \min_{i\neq j} \left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|_\infty > \varepsilon\Big\}.$$

Theorem

If f is a non-constant continuous function, for any $\min(f) < a < \max(f)$,

$$\mathcal{N}\Big(\{f=a\},\varepsilon\Big) \gtrsim \frac{1}{\varepsilon^{d-1}}$$

as $\varepsilon \to 0$.

Not surprising, the level set is defined by a single equation in d unknowns.

A hard problem

Definition

The packing number of a non-empty set E is

$$\mathcal{N}(E,\varepsilon) := \sup\left\{k \in \mathbb{N} : \exists\, \boldsymbol{x}_1,\ldots,\boldsymbol{x}_k \in E,\ \min_{i\neq j}\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|_\infty > \varepsilon\right\}.$$

Theorem

If f is a non-constant continuous function, for any $\min(f) < a < \max(f)$,

$$\mathcal{N}\Big(\{f = a\},\varepsilon\Big) \gtrsim \frac{1}{\varepsilon^{d-1}}$$

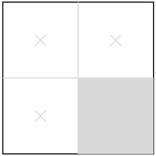
as $\varepsilon \to 0$.

Not surprising, the level set is defined by a single equation in d unknowns.

A general solution

Bisect and Approximate (BA)

1. Bisect the current family of cells.
2. Query f at some point(s) in each new cell.
3. Compute a local approximator g_C of f on each cell C.
4. Remove a cell C if $|g_C(\boldsymbol{x}) - a|$ is large for all $\boldsymbol{x} \in C$.



Theorem

If the g_C's are "accurate approximations" of f on the C's,

$$\text{sample complexity of BA} \lesssim \sum_{i=1}^{i(\varepsilon)} \mathcal{N}\Big(\{|f - a| \le c_i\}, d_i\Big)$$

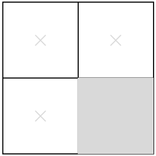
where $i(\varepsilon) \sim \log(1/\varepsilon)$, $c_1 > c_2 > \ldots$, $d_1 > d_2 > \ldots$ depend on the g_C's and their error bounds.

Instance-dependent bound (depends on f, not only on smoothness class).

A general solution

Bisect and Approximate (BA)

1. Bisect the current family of cells.
2. Query f at some point(s) in each new cell.
3. Compute a local approximator g_C of f on each cell C.
4. Remove a cell C if $|g_C(\boldsymbol{x}) - a|$ is large for all $\boldsymbol{x} \in C$.



Theorem

If the g_C's are "accurate approximations" of f on the C's,

$$\text{sample complexity of BA} \lesssim \sum_{i=1}^{I(\varepsilon)} \mathcal{N}\Big(\{|f - a| \leq c_i\}, d_i\Big)$$

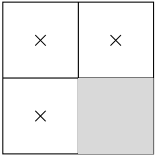
where $I(\varepsilon) \sim \log(1/\varepsilon)$, $c_1 > c_2 > \dots$, $d_1 > d_2 > \dots$ depend on the g_C's and their error bounds.

Instance-dependent bound (depends on f, not only on smoothness class).

A general solution

Bisect and Approximate (BA)

1. Bisect the current family of cells.
2. Query f at some point(s) in each new cell.
3. Compute a local approximator g_C of f on each cell C.
4. Remove a cell C if $|g_C(\boldsymbol{x}) - a|$ is large for all $\boldsymbol{x} \in C$.



Theorem

If the g_C's are "accurate approximations" of f on the C's,

$$\text{sample complexity of BA} \lesssim \sum_{i=1}^{I(\varepsilon)} \mathcal{N}\Big(\{|f-a| \le c_i\}, d_i\Big)$$

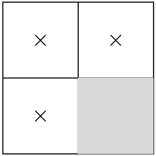
where $I(\varepsilon) \sim \log(1/\varepsilon)$, $c_1 > c_2 > \ldots$, $d_1 > d_2 > \ldots$ depend on the g_C's and their error bounds.

Instance-dependent bound (depends on f, not only on smoothness class).

A general solution

Bisect and Approximate (BA)

1. Bisect the current family of cells.
2. Query f at some point(s) in each new cell.
3. Compute a local approximator g_C of f on each cell C.
4. Remove a cell C if $|g_C(\boldsymbol{x}) - a|$ is large for all $\boldsymbol{x} \in C$.



Theorem

If the g_C's are "accurate approximations" of f on the C's,

$$\text{sample complexity of BA} \lesssim \sum_{i=1}^{I(\varepsilon)} \mathcal{N}\Big(\{|f - a| \le c_i\}, d_i\Big)$$

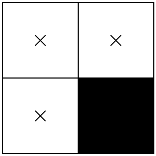
where $I(\varepsilon) \sim \log(1/\varepsilon)$, $c_1 > c_2 > \ldots$, $d_1 > d_2 > \ldots$ depend on the g_C's and their error bounds.

Instance-dependent bound (depends on f, not only on smoothness class).

A general solution

Bisect and Approximate (BA)

1. Bisect the current family of cells.
2. Query f at some point(s) in each new cell.
3. Compute a local approximator g_C of f on each cell C.
4. Remove a cell C if $|g_C(\boldsymbol{x}) - a|$ is large for all $\boldsymbol{x} \in C$.



Theorem

If the g_C's are "accurate approximations" of f on the C's,

$$\text{sample complexity of BA} \lesssim \sum_{i=1}^{i(\varepsilon)} \mathcal{N}\Big(\{|f-a| \leq c_i\}, d_i\Big)$$

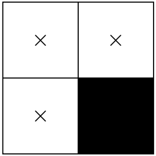
where $i(\varepsilon) \sim \log(1/\varepsilon)$, $c_1 > c_2 > \dots$, $d_1 > d_2 > \dots$ depend on the g_C's and their error bounds.

Instance-dependent bound (depends on f, not only on smoothness class).

A general solution

Bisect and Approximate (BA)

1. Bisect the current family of cells.
2. Query f at some point(s) in each new cell.
3. Compute a local approximator g_C of f on each cell C.
4. Remove a cell C if $|g_C(\boldsymbol{x}) - a|$ is large for all $\boldsymbol{x} \in C$.



Theorem

If the g_C's are "accurate approximations" of f on the C's,

$$\text{sample complexity of BA} \lesssim \sum_{i=1}^{i(\varepsilon)} \mathcal{N}\Big(\{|f - a| \leq c_i\}, d_i\Big)$$

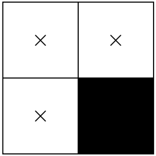
where $i(\varepsilon) \sim \log(1/\varepsilon)$, $c_1 > c_2 > \dots$, $d_1 > d_2 > \dots$ depend on the g_C's and their error bounds.

Instance-dependent bound (depends on f, not only on smoothness class).

A general solution

Bisect and Approximate (BA)

1. Bisect the current family of cells.
2. Query f at some point(s) in each new cell.
3. Compute a local approximator g_C of f on each cell C.
4. Remove a cell C if $|g_C(\boldsymbol{x}) - a|$ is large for all $\boldsymbol{x} \in C$.



Theorem

If the g_C's are "accurate approximations" of f on the C's,

$$\text{sample complexity of BA} \lesssim \sum_{i=1}^{i(\varepsilon)} \mathcal{N}\Big(\{|f - a| \leq c_i\}, d_i\Big)$$

where $i(\varepsilon) \sim \log(1/\varepsilon)$, $c_1 > c_2 > \ldots$, $d_1 > d_2 > \ldots$ depend on the g_C's and their error bounds.

Instance-dependent bound (depends on f, not only on smoothness class).

Sketch of proof for Lipschitz functions

- Query f only at center of C. Predict constantly by observed value.
- Prediction error $\leq \mathrm{Const}/2^i$ after i bisections.
- $\Longrightarrow$ Level set estimation error $\leq \varepsilon$ when $\mathrm{Const}/2^i \leq \varepsilon \Longrightarrow$ after $\mathrm{Const}\log(1/\varepsilon)$ bisections.
- Query points at bisection i are in $\left\{\boldsymbol{x} : |f(\boldsymbol{x}) - a| \leq \mathrm{Const}/2^i\right\}$ and are a packing with radius $\mathrm{Const}/2^i$.
- Hence

$$\text{Total number queried points} \leq \sum_{i=1}^{\mathrm{const}\log(1/\varepsilon)} \mathcal{N}\left(\left\{|f-a| \leq \frac{\mathrm{Const}}{2^i}\right\}, \frac{\mathrm{Const}}{2^i}\right).$$

Sketch of proof for Lipschitz functions

- Query f only at center of C. Predict constantly by observed value.
- Prediction error $\leq \mathrm{Const}/2^i$ after i bisections.
- $\Longrightarrow$ Level set estimation error $\leq \varepsilon$ when $\mathrm{Const}/2^i \leq \varepsilon \Longrightarrow$ after $\mathrm{Const} \log(1/\varepsilon)$ bisections.
- Query points at bisection i are in $\left\{ \boldsymbol{x} : |f(\boldsymbol{x}) - a| \leq \mathrm{Const}/2^i \right\}$ and are a packing with radius $\mathrm{Const}/2^i$.
- Hence

$$\text{Total number queried points} \leq \sum_{i=1}^{\mathrm{const}\log(1/\varepsilon)} \mathcal{N}\left(\left\{ |f - a| \leq \frac{\mathrm{Const}}{2^i} \right\}, \frac{\mathrm{Const}}{2^i} \right).$$

Sketch of proof for Lipschitz functions

- Query f only at center of C. Predict constantly by observed value.
- Prediction error $\leq \mathrm{Const}/2^i$ after i bisections.
- $\Longrightarrow$ Level set estimation error $\leq \varepsilon$ when $\mathrm{Const}/2^i \leq \varepsilon \Longrightarrow$ after $\mathrm{Const}\log(1/\varepsilon)$ bisections.
- Query points at bisection i are in $\{\boldsymbol{x} : |f(\boldsymbol{x}) - a| \leq \mathrm{Const}/2^i\}$ and are a packing with radius $\mathrm{Const}/2^i$.
- Hence

$$\text{Total number queried points} \leq \sum_{i=1}^{\mathrm{const}\log(1/\varepsilon)} \mathcal{N}\left(\left\{|f - a| \leq \frac{\mathrm{Const}}{2^i}\right\}, \frac{\mathrm{Const}}{2^i}\right).$$

Sketch of proof for Lipschitz functions

- Query f only at center of C. Predict constantly by observed value.
- Prediction error $\leq \mathrm{Const}/2^i$ after i bisections.
- $\Longrightarrow$ Level set estimation error $\leq \varepsilon$ when $\mathrm{Const}/2^i \leq \varepsilon \Longrightarrow$ after $\mathrm{Const}\log(1/\varepsilon)$ bisections.
- Query points at bisection i are in $\left\{\boldsymbol{x} : |f(\boldsymbol{x}) - a| \leq \mathrm{Const}/2^i\right\}$ and are a packing with radius $\mathrm{Const}/2^i$.
- Hence

$$\text{Total number queried points} \leq \sum_{i=1}^{\mathrm{const}\log(1/\varepsilon)} \mathcal{N}\left(\left\{|f - a| \leq \frac{\mathrm{Const}}{2^i}\right\}, \frac{\mathrm{Const}}{2^i}\right).$$

Sketch of proof for Lipschitz functions

- Query f only at center of C. Predict constantly by observed value.
- Prediction error $\leq \mathrm{Const}/2^i$ after i bisections.
- $\Longrightarrow$ Level set estimation error $\leq \varepsilon$ when $\mathrm{Const}/2^i \leq \varepsilon \Longrightarrow$ after $\mathrm{Const} \log(1/\varepsilon)$ bisections.
- Query points at bisection i are in $\left\{ \boldsymbol{x} : |f(\boldsymbol{x}) - a| \leq \mathrm{Const}/2^i \right\}$ and are a packing with radius $\mathrm{Const}/2^i$.
- Hence

$$\text{Total number queried points} \leq \sum_{i=1}^{\mathrm{const} \log(1/\varepsilon)} \mathcal{N}\left(\{ |f - a| \leq \frac{\mathrm{Const}}{2^i} \}, \frac{\mathrm{Const}}{2^i} \right).$$

Consequence for γ-Hölder functions

γ-Hölder functions

f is γ-Hölder : $\left|f(\boldsymbol{x}) - f(\boldsymbol{y})\right| \leq c\,\|\boldsymbol{x} - \boldsymbol{y}\|^{\gamma}$, with $\gamma \in (0, 1]$.

Local approximator for BAH

- For a cell C (hypercube), we query the center and take the local approximator g_C as constant.
- The error of g_C on C is $\lesssim \mathrm{Diam}(C)^{\gamma}$.

Theorem (upper and lower bound)

The worst-case optimal sample complexity is attained by BAH and

$$\text{sample complexity of BAH} \lesssim \frac{1}{\varepsilon^{d/\gamma}}.$$

For lower bound counter example functions are "flat + local bump" (classical).

Consequence for γ-Hölder functions

γ-Hölder functions

f is γ-Hölder : $\left|f(\boldsymbol{x}) - f(\boldsymbol{y})\right| \leq c \left\|\boldsymbol{x} - \boldsymbol{y}\right\|^{\gamma}$, with $\gamma \in (0, 1]$.

Local approximator for BAH

- For a cell C (hypercube), we query the center and take the local approximator g_C as constant.
- The error of g_C on C is $\lesssim \mathrm{Diam}(C)^{\gamma}$.

Theorem (upper and lower bound)

The worst-case optimal sample complexity is attained by BAH and

$$\text{sample complexity of BAH} \lesssim \frac{1}{\varepsilon^{d/\gamma}}.$$

For lower bound counter example functions are "flat + local bump" (classical).

Consequence for γ-Hölder functions

γ-Hölder functions

f is γ-Hölder : $\left|f(\boldsymbol{x}) - f(\boldsymbol{y})\right| \leq c \left\|\boldsymbol{x} - \boldsymbol{y}\right\|^{\gamma}$, with $\gamma \in (0, 1]$.

Local approximator for BAH

- For a cell C (hypercube), we query the center and take the local approximator g_C as constant.
- The error of g_C on C is $\lesssim \mathrm{Diam}(C)^{\gamma}$.

Theorem (upper and lower bound)

The worst-case optimal sample complexity is attained by BAH and

$$\text{sample complexity of BAH} \lesssim \frac{1}{\varepsilon^{d/\gamma}}.$$

For lower bound counter example functions are “flat + local bump” (classical).

Consequence for functions with γ_1-Hölder gradients

Functions with γ_1-Hölder gradients

∇f is γ_1-Hölder.

Local approximator for BAG

- For a cell C (hypercube), we query the 2^d vertices and take the local approximator g_C as multilinear interpolating.
- The error of g_C on C is $\lesssim \mathrm{Diam}(C)^{1+\gamma_1}$.

Theorem (upper and lower bound)

The worst-case optimal sample complexity is attained by BAG and

$$\text{sample complexity of BAG} \lesssim \frac{1}{\varepsilon^{d/(1+\gamma_1)}}.$$

For lower bound counter example functions are “flat + local bump” (classical).

Consequence for functions with γ_1-Hölder gradients

Functions with γ_1-Hölder gradients

∇f is γ_1-Hölder.

Local approximator for BAG

- For a cell C (hypercube), we query the 2^d vertices and take the local approximator g_C as multilinear interpolating.
- The error of g_C on C is $\lesssim \mathrm{Diam}(C)^{1+\gamma_1}$.

Theorem (upper and lower bound)

The worst-case optimal sample complexity is attained by BAG and

$$\text{sample complexity of BAG} \lesssim \frac{1}{\varepsilon^{d/(1+\gamma_1)}}.$$

For lower bound counter example functions are "flat + local bump" (classical).

Consequence for functions with γ_1-Hölder gradients

Functions with γ_1-Hölder gradients

∇f is γ_1-Hölder.

Local approximator for BAG

- For a cell C (hypercube), we query the 2^d vertices and take the local approximator g_C as multilinear interpolating.
- The error of g_C on C is $\lesssim \mathrm{Diam}(C)^{1+\gamma_1}$.

Theorem (upper and lower bound)

The worst-case optimal sample complexity is attained by BAG and

$$\text{sample complexity of BAG} \lesssim \frac{1}{\varepsilon^{d/(1+\gamma_1)}}.$$

For lower bound counter example functions are “flat + local bump” (classical).

Convexity helps a little : $d \mapsto d - 1$

Theorem

When f has γ_1-Hölder gradient and is convex (+ quantitative conditions), then the worst-case optimal sample complexity is attained by BAG and

$$\text{sample complexity of BAG} \lesssim \frac{1}{\varepsilon^{(d-1)/(1+\gamma_1)}}.$$

Follows from geometric arguments on level sets of convex functions.

Theorem

If f is convex (+ quantitative conditions), there exists a constant $C^* > 0$ such that

$$\forall r \in (0,1)\,,\ \mathcal{N}\Big(\{|f - a| \le r\},\ r\Big) \le C^* \left(\frac{1}{r}\right)^{d-1}.$$

Convexity helps a little : $d \mapsto d-1$

Theorem

When f has γ_1-Hölder gradient and is convex (+ quantitative conditions), then the worst-case optimal sample complexity is attained by BAG and

$$\text{sample complexity of BAG} \lesssim \frac{1}{\varepsilon^{(d-1)/(1+\gamma_1)}}.$$

Follows from geometric arguments on level sets of convex functions.

Theorem

If f is convex (+ quantitative conditions), there exists a constant $C^* > 0$ such that

$$\forall r \in (0,1)\ ,\ \mathcal{N}\Big(\{|f-a| \leq r\},\ r\Big) \leq C^* \left(\frac{1}{r}\right)^{d-1}.$$

Setting

Goal

Determine and **certify** a near-maximizer for a black-box function $f \colon \mathcal{X} \subset \mathbb{R}^d \to \mathbb{R}$ that is L-Lipschitz w.r.t. a norm $\|\cdot\|$.

L is a known upper bound on the smallest possible Lipschitz constant $\mathrm{Lip}(f)$.

Certified Algorithm

For $n = 1, 2, \ldots$

1. query $\boldsymbol{x}_n \in [0,1]^d$,
2. observe $f(\boldsymbol{x}_n)$,
3. output a recommendation $\boldsymbol{x}_n^\star \in [0,1]^d$, with the goal of minimizing $\max(f) - f(\boldsymbol{x}_n^\star)$,
4. output an error certificate $\xi_n \geq 0$ such that $\max(f) - f(\boldsymbol{x}_n^\star) \leq \xi_n$.

Setting

Goal

Determine and **certify** a near-maximizer for a black-box function $f\colon \mathcal{X} \subset \mathbb{R}^d \to \mathbb{R}$ that is L-Lipschitz w.r.t. a norm $\|\cdot\|$.

L is a known upper bound on the smallest possible Lipschitz constant $\mathrm{Lip}(f)$.

Certified Algorithm

For $n = 1, 2, \ldots$

1. query $\boldsymbol{x}_n \in [0,1]^d$,
2. observe $f(\boldsymbol{x}_n)$,
3. output a recommendation $\boldsymbol{x}_n^\star \in [0,1]^d$, with the goal of minimizing $\max(f) - f(\boldsymbol{x}_n^\star)$,
4. output an error certificate $\xi_n \geq 0$ such that $\max(f) - f(\boldsymbol{x}_n^\star) \leq \xi_n$.

Setting

Goal

Determine and **certify** a near-maximizer for a black-box function $f\colon \mathcal{X} \subset \mathbb{R}^d \to \mathbb{R}$ that is L-Lipschitz w.r.t. a norm $\|\cdot\|$.

L is a known upper bound on the smallest possible Lipschitz constant $\mathrm{Lip}(f)$.

Certified Algorithm

For $n = 1, 2, \ldots$

1. query $\boldsymbol{x}_n \in [0, 1]^d$,
2. observe $f(\boldsymbol{x}_n)$,
3. output a recommendation $\boldsymbol{x}_n^\star \in [0, 1]^d$, with the goal of minimizing $\max(f) - f(\boldsymbol{x}_n^\star)$,
4. output an error certificate $\xi_n \geq 0$ such that $\max(f) - f(\boldsymbol{x}_n^\star) \leq \xi_n$.

Setting

Goal

Determine and **certify** a near-maximizer for a black-box function $f\colon \mathcal{X} \subset \mathbb{R}^d \to \mathbb{R}$ that is L-Lipschitz w.r.t. a norm $\|\cdot\|$.

L is a known upper bound on the smallest possible Lipschitz constant $\mathrm{Lip}(f)$.

Certified Algorithm

For $n = 1, 2, \ldots$

1. query $\boldsymbol{x}_n \in [0, 1]^d$,
2. observe $f(\boldsymbol{x}_n)$,
3. output a recommendation $\boldsymbol{x}_n^\star \in [0, 1]^d$, with the goal of minimizing $\max(f) - f(\boldsymbol{x}_n^\star)$,
4. output an error certificate $\xi_n \geq 0$ such that $\max(f) - f(\boldsymbol{x}_n^\star) \leq \xi_n$.

Setting

Goal

Determine and **certify** a near-maximizer for a black-box function $f \colon \mathcal{X} \subset \mathbb{R}^d \to \mathbb{R}$ that is L-Lipschitz w.r.t. a norm $\|\cdot\|$.

L is a known upper bound on the smallest possible Lipschitz constant $\mathrm{Lip}(f)$.

Certified Algorithm

For $n = 1, 2, \ldots$

1. query $\boldsymbol{x}_n \in [0, 1]^d$,
2. observe $f(\boldsymbol{x}_n)$,
3. output a recommendation $\boldsymbol{x}_n^\star \in [0, 1]^d$, with the goal of minimizing $\max(f) - f(\boldsymbol{x}_n^\star)$,
4. output an error certificate $\xi_n \geq 0$ such that $\max(f) - f(\boldsymbol{x}_n^\star) \leq \xi_n$.

Goal

Determine and **certify** a near-maximizer for a black-box function $f\colon \mathcal{X} \subset \mathbb{R}^d \to \mathbb{R}$ that is L-Lipschitz w.r.t. a norm $\|\cdot\|$.

L is a known upper bound on the smallest possible Lipschitz constant $\mathrm{Lip}(f)$.

Certified Algorithm

For $n = 1, 2, \ldots$

1. query $\boldsymbol{x}_n \in [0, 1]^d$,
2. observe $f(\boldsymbol{x}_n)$,
3. output a recommendation $\boldsymbol{x}_n^\star \in [0, 1]^d$, with the goal of minimizing $\max(f) - f(\boldsymbol{x}_n^\star)$,
4. output an error certificate $\xi_n \geq 0$ such that $\max(f) - f(\boldsymbol{x}_n^\star) \leq \xi_n$.

Setting

Goal

Determine and **certify** a near-maximizer for a black-box function $f\colon \mathcal{X} \subset \mathbb{R}^d \to \mathbb{R}$ that is L-Lipschitz w.r.t. a norm $\|\cdot\|$.

L is a known upper bound on the smallest possible Lipschitz constant $\mathrm{Lip}(f)$.

Certified Algorithm

For $n = 1, 2, \ldots$

1. query $\boldsymbol{x}_n \in [0,1]^d$,
2. observe $f(\boldsymbol{x}_n)$,
3. output a recommendation $\boldsymbol{x}_n^\star \in [0,1]^d$, with the goal of minimizing $\max(f) - f(\boldsymbol{x}_n^\star)$,
4. output an error certificate $\xi_n \geq 0$ such that $\max(f) - f(\boldsymbol{x}_n^\star) \leq \xi_n$.

Upper bound

$\mathcal{F}_L$ is the set of all L-Lipschitz functions with $\mathrm{Lip}(f) < L$.

Sample complexity of an algorithm A

$\sigma(A, f, \varepsilon)$ is the smallest n such that $\xi_n \leq \varepsilon$.

We introduce the algorithm c.DOO (certified Deterministic Optimistic Optimization), extending DOO from [Munos, 2011].

Proposition

c.DOO is a certified algorithm and for all $f \in \mathcal{F}_L$,

$$\sigma(\text{c.DOO}, f, \varepsilon) \lesssim S_{\mathrm{C}}(f, \varepsilon)$$

where

$$S_{\mathrm{C}}(f, \varepsilon) := \mathcal{N}\left(\{\boldsymbol{x}; \max f - f(\boldsymbol{x}) \leq \varepsilon\}, \frac{\varepsilon}{L}\right) + \sum_{k=1}^{\mathrm{Const}\log(1/\varepsilon)} \mathcal{N}\left(\{\boldsymbol{x}; \varepsilon_k < \max f - f(\boldsymbol{x}) \leq \varepsilon_{k-1}\}, \frac{\varepsilon_k}{L}\right)$$

with $\varepsilon_{\mathrm{Const}\log(1/\varepsilon)} = \varepsilon$, $\varepsilon_{\mathrm{Const}\log(1/\varepsilon)-1} > \varepsilon$, and then $\varepsilon_{i-1} = 2\varepsilon_i$, $i = \mathrm{Const}\log(1/\varepsilon) - 1, \dots, 1$ and with $\epsilon_0 = L$.

Upper bound

$\mathcal{F}_L$ is the set of all L-Lipschitz functions with $\mathrm{Lip}(f) < L$.

Sample complexity of an algorithm A

$\sigma(A, f, \varepsilon)$ is the smallest n such that $\xi_n \leq \varepsilon$.

We introduce the algorithm c.DOO (certified Deterministic Optimistic Optimization), extending DOO from [Munos, 2011].

Proposition

c.DOO is a certified algorithm and for all $f \in \mathcal{F}_L$,

$$\sigma(\text{c.DOO}, f, \varepsilon) \lesssim S_{\mathrm{C}}(f, \varepsilon)$$

where

$$S_{\mathrm{C}}(f,\varepsilon) := \mathcal{N}\left(\{\boldsymbol{x}; \max f - f(\boldsymbol{x}) \leq \varepsilon\}, \frac{\varepsilon}{L}\right) + \sum_{k=1}^{\mathrm{Const}\log(1/\varepsilon)} \mathcal{N}\left(\{\boldsymbol{x}; \varepsilon_k < \max f - f(\boldsymbol{x}) \leq \varepsilon_{k-1}\}, \frac{\varepsilon_k}{L}\right)$$

with $\varepsilon_{\mathrm{Const}\log(1/\varepsilon)} = \varepsilon$, $\varepsilon_{\mathrm{Const}\log(1/\varepsilon)-1} > \varepsilon$, and then $\varepsilon_{i-1} = 2\varepsilon_i$, $i = \mathrm{Const}\log(1/\varepsilon) - 1, \ldots, 1$ and with $\epsilon_0 = L$.

Upper bound

$\mathcal{F}_L$ is the set of all L-Lipschitz functions with $\mathrm{Lip}(f) < L$.

Sample complexity of an algorithm A

$\sigma(A, f, \varepsilon)$ is the smallest n such that $\xi_n \leq \varepsilon$.

We introduce the algorithm c.DOO (certified Deterministic Optimistic Optimization), extending DOO from [Munos, 2011].

Proposition

c.DOO is a certified algorithm and for all $f \in \mathcal{F}_L$,

$$\sigma(\text{c.DOO}, f, \varepsilon) \lesssim S_{\mathrm{C}}(f, \varepsilon)$$

where

$$S_{\mathrm{C}}(f, \varepsilon) := \mathcal{N}\left(\{\boldsymbol{x}; \max f - f(\boldsymbol{x}) \leq \varepsilon\}, \frac{\varepsilon}{L}\right) + \sum_{k=1}^{\mathrm{Const}\log(1/\varepsilon)} \mathcal{N}\left(\{\boldsymbol{x}; \varepsilon_k < \max f - f(\boldsymbol{x}) \leq \varepsilon_{k-1}\}, \frac{\varepsilon_k}{L}\right)$$

with $\varepsilon_{\mathrm{Const}\log(1/\varepsilon)} = \varepsilon$, $\varepsilon_{\mathrm{Const}\log(1/\varepsilon)-1} > \varepsilon$, and then $\varepsilon_{i-1} = 2\varepsilon_i$, $i = \mathrm{Const}\log(1/\varepsilon) - 1, \dots, 1$ and with $\epsilon_0 = L$.

Upper bound

$\mathcal{F}_L$ is the set of all L-Lipschitz functions with $\mathrm{Lip}(f) < L$.

Sample complexity of an algorithm A

$\sigma(A, f, \varepsilon)$ is the smallest n such that $\xi_n \leq \varepsilon$.

We introduce the algorithm c.DOO (certified Deterministic Optimistic Optimization), extending DOO from [Munos, 2011].

Proposition

c.DOO is a certified algorithm and for all $f \in \mathcal{F}_L$,

$$\sigma(\text{c.DOO}, f, \varepsilon) \lesssim S_{\mathrm{C}}(f, \varepsilon)$$

where

$$S_{\mathrm{C}}(f, \varepsilon) := \mathcal{N}\left(\{\boldsymbol{x}; \max f - f(\boldsymbol{x}) \leq \varepsilon\}, \frac{\varepsilon}{L}\right) + \sum_{k=1}^{\mathrm{Const}\,\log(1/\varepsilon)} \mathcal{N}\left(\{\boldsymbol{x}; \varepsilon_k < \max f - f(\boldsymbol{x}) \leq \varepsilon_{k-1}\}, \frac{\varepsilon_k}{L}\right)$$

with $\varepsilon_{\mathrm{Const}\,\log(1/\varepsilon)} = \varepsilon$, $\varepsilon_{\mathrm{Const}\,\log(1/\varepsilon)-1} > \varepsilon$, and then $\varepsilon_{i-1} = 2\varepsilon_i$, $i = \mathrm{Const}\,\log(1/\varepsilon) - 1, \ldots, 1$ and with $\epsilon_0 = L$.

Integral characterization and lower bound

Theorem

Under a mild assumption on $\mathcal{X}$, $S_{\mathrm{C}}(f,\varepsilon) \sim \int_{\mathcal{X}} \frac{\mathrm{d}\boldsymbol{x}}{\big(f(\boldsymbol{x}^\star) - f(\boldsymbol{x}) + \varepsilon\big)^d}$.

$\Longrightarrow$ The constant function is the hardest for certified optimization.

Theorem

For all $f \in \mathcal{F}_L$, the sample complexity of any certified algorithm A satisfies

$$\sigma(A,f,\varepsilon) \geq \frac{\mathrm{Const}\big(1 - \mathrm{Lip}(f)/L\big)^d \cdot S_{\mathrm{C}}(f,\varepsilon)}{\log(1/\varepsilon)}.$$

This instance-dependent lower bound makes sense only in the certified setting.

Extends the one-dimensional analysis of [Hansen et al., 1991].

Integral characterization and lower bound

Theorem

Under a mild assumption on $\mathcal{X}$, $S_{\mathrm{C}}(f,\varepsilon) \sim \int_{\mathcal{X}} \frac{\mathrm{d}\boldsymbol{x}}{\big(f(\boldsymbol{x}^\star) - f(\boldsymbol{x}) + \varepsilon\big)^d}$.

$\Longrightarrow$ The constant function is the hardest for certified optimization.

Theorem

For all $f \in \mathcal{F}_L$, the sample complexity of any certified algorithm A satisfies

$$\sigma(A, f, \varepsilon) \geq \frac{\mathrm{Const}\big(1 - \mathrm{Lip}(f)/L\big)^d \cdot S_{\mathrm{C}}(f,\varepsilon)}{\log(1/\varepsilon)}.$$

This instance-dependent lower bound makes sense only in the certified setting.

Extends the one-dimensional analysis of [Hansen et al., 1991].

Integral characterization and lower bound

Theorem

Under a mild assumption on $\mathcal{X}$, $S_{\mathrm{C}}(f, \varepsilon) \sim \int_{\mathcal{X}} \frac{\mathrm{d}\boldsymbol{x}}{\left(f(\boldsymbol{x}^\star) - f(\boldsymbol{x}) + \varepsilon\right)^d}$.

$\Longrightarrow$ The constant function is the hardest for certified optimization.

Theorem

For all $f \in \mathcal{F}_L$, the sample complexity of any certified algorithm A satisfies

$$\sigma(A, f, \varepsilon) \geq \frac{\mathrm{Const}\left(1 - \mathrm{Lip}(f)/L\right)^d \cdot S_{\mathrm{C}}(f, \varepsilon)}{\log(1/\varepsilon)}.$$

This instance-dependent lower bound makes sense only in the certified setting.

Extends the one-dimensional analysis of [Hansen et al., 1991].

Thank you for your attention !

Azzimonti, D., Ginsbourger, D., Chevalier, C., Bect, J., and Richet, Y. (2021).
Adaptive design of experiments for conservative estimation of excursion sets.
Technometrics, 63(1) :13–26.

Bect, J., Bachoc, F., and Ginsbourger, D. (2019).
A supermartingale approach to Gaussian process based sequential design of experiments.
Bernoulli, 25(4A) :2883–2919.

Bubeck, S., Munos, R., Stoltz, G., and Szepesvari, C. (2011).
X-armed bandits.
Journal of Machine Learning Research, 12 :1655–1695.

Bull, A. D. (2011).
Convergence rates of efficient global optimization algorithms.
Journal of Machine Learning Research, 12 :2879–2904.

Chevalier, C., Bect, J., Ginsbourger, D., Vazquez, E., Picheny, V., and Richet, Y. (2014).
Fast parallel kriging-based stepwise uncertainty reduction with application to the identification of an excursion set.
Technometrics, 56(4) :455–465.

de Freitas, N., Smola, A. J., and Zoghi, M. (2012).
Exponential regret bounds for Gaussian process bandits with deterministic observations.
In *Proceedings of the 29th International Coference on International Conference on Machine Learning*.

Gotovos, A., Casati, N., Hitz, G., and Krause, A. (2013).
Active learning for level set estimation.
In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 1344–1350. AAAI Press.

Hansen, P., Jaumard, B., and Lu, S.-H. (1991).
On the number of iterations of Piyavskii's global optimization algorithm.
Mathematics of Operations Research, 16(2) :334–350.

Jones, D., Schonlau, M., and Welch, W. (1998).
Efficient global optimization of expensive black box functions.
Journal of Global Optimization, 13 :455–492.

Munos, R. (2011).
Optimistic optimization of a deterministic function without the knowledge of its smoothness.
In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA. Curran Associates Inc.

Srinivas, N., A., K., Kakade, S., and Seeger, M. (2012).
Information-theoretic regret bounds for Gaussian process optimization in the bandit setting.
IEEE Transactions on Information Theory, 58 :3250–3265.

Vazquez, E. and Bect, J. (2010).
Convergence properties of the expected improvement algorithm with fixed mean and covariance functions.
Journal of Statistical Planning and Inference, 140(11) :3088–3095.