
The Sample Complexity of Level Set Approximation

François Bachoc

*University Paul Sabatier
Institut de Mathématiques de Toulouse
francois.bachoc@math.univ-toulouse.fr*

Tommaso Cesari

*Toulouse School of Economics
tommaso.cesari@tse-fr.eu*

Sébastien Gerchinovitz

*IRT Saint Exupéry
Institut de Mathématiques de Toulouse
sebastien.gerchinovitz@irt-saintexupery.com*

Abstract

We study the problem of approximating the level set of an unknown function by sequentially querying its values. We introduce a family of algorithms called Bisect and Approximate through which we reduce the level set approximation problem to a local function approximation problem. We then show how this approach leads to rate-optimal sample complexity guarantees for Hölder functions, and we investigate how such rates improve when additional smoothness or other structural assumptions hold true.

1 INTRODUCTION

Let $f: [0, 1]^d \rightarrow \mathbb{R}$ be any function. For $a \in \mathbb{R}$, we consider the problem of finding the level set

$$\{f = a\} \stackrel{\text{def}}{=} \{\mathbf{x} \in [0, 1]^d : f(\mathbf{x}) = a\}.$$

Setting: Sequential Black-Box Evaluation.

We study the case in which f is black-box, i.e., except for some *a priori* knowledge on its smoothness, we can only access f by sequentially querying its values at a sequence $\mathbf{x}_1, \mathbf{x}_2, \dots \in [0, 1]^d$ of points of our choice (Online Protocol 1). At every round $n \geq 1$, the query point \mathbf{x}_n can be chosen as a deterministic function of the values $f(\mathbf{x}_1), \dots, f(\mathbf{x}_{n-1})$ observed so far. At the end of round n , the learner outputs a subset S_n of $[0, 1]^d$ with the goal of approximating the level set $\{f = a\}$.

Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130. Copyright 2021 by the author(s).

Online Protocol 1: Deterministic Scheme

```
1 for  $n = 1, 2, \dots$  do  
2   pick the next query point  $\mathbf{x}_n \in [0, 1]^d$   
3   observe the value  $f(\mathbf{x}_n)$   
4   output an approximating set  $S_n \subseteq [0, 1]^d$ 
```

The problem of identifying the level set $\{f = a\}$ of a black-box function arises often in practice. In particular, this problem is closely related to excursion set estimation (also called failure domain estimation), where the goal is to estimate $\{f \geq a\}$.¹ Level-set identification and failure domain estimation are relevant to the field of computer experiments and uncertainty quantification, where $f(x)$ provides the output of a complex computer model for some input parameter x (Sacks et al., 1989; Santner et al., 2003). Typical fields of applications are nuclear engineering (Chevalier et al., 2014), coastal flooding (Azzimonti et al., 2020) and network systems (Ranjan et al., 2008). Level set identification is also relevant when $f(x)$ corresponds to natural data (Rahimi et al., 2004; Galland et al., 2004). In many such situations, f is so complex that it is considered black-box.

A typical example of a real use-case of interest is the Bombardier research aircraft configuration (Priem et al., 2020). Here, geometry parameters of an aircraft wing can be selected. Any choice of these parameters yields a corresponding maximum take-off weight output, which is obtained by a costly computational fluid dynamics simulation. From the setting of Priem et al. (2020), one could for instance tackle the problem of estimating the set of all $\mathbf{x} \in [0, 1]^4$, where \mathbf{x} corresponds to the variables *wing span*, *wing leading edge sweep*, *wing break location* and *wingtip*

¹As it will become apparent later, our techniques for estimating level sets can be adapted for sub/superlevel set approximation straightforwardly, whilst retaining the same sample complexity guarantees (see Footnote 4).

chord (see Table 3 in Priem et al. 2020), for which $f(\mathbf{x}) = a$ for some prescribed value $a > 0$ of the maximum take-off weight. Furthermore, by setting more or less input parameters as active or inactive in Table 3 in Priem et al. (2020), a series of level set estimation problems can be obtained, from dimension 1 to dimension 18.

Learning Goal. There exist several ways to compare the estimators S_n and the level set $\{f = a\}$. A first possibility is to use metrics or pseudometrics $\rho(A, B)$ between sets $A, B \subseteq [0, 1]^d$, such as the Hausdorff distance or the volume of the symmetric difference (e.g., Tsybakov 1997). However a small value of $\rho(S_n, \{f = a\})$ does not imply that S_n contains the whole set $\{f = a\}$, nor—in the case of the volume of the symmetric difference—that $f(x) \approx a$ for all $\mathbf{x} \in S_n$. In practice, we might fail to identify *all* critical states of a given system, or raise unnecessary false alarms.

In this paper, we therefore consider an alternative (new) way of quantifying our performance. For any accuracy $\varepsilon > 0$, denote by

$$\{|f - a| \leq \varepsilon\} \stackrel{\text{def}}{=} \{\mathbf{x} \in [0, 1]^d : |f(\mathbf{x}) - a| \leq \varepsilon\}$$

the *inflated* level set at scale ε . We will focus on algorithms whose outputs S_n are ε -approximations of $\{f = a\}$, as defined below.

Definition 1 (ε -approximation of a level set). We say that a set $S \subseteq [0, 1]^d$ is an ε -*approximation* of the level set $\{f = a\}$ if and only if it contains $\{f = a\}$ while consisting only of points at which f is at most ε -away from a , i.e.,

$$\{f = a\} \subseteq S \subseteq \{|f - a| \leq \varepsilon\}. \quad (1)$$

The main mathematical problem we address is that of determining the *sample complexity* of level set approximation, that is, the minimum number of evaluations of f after which S_n is an ε -approximation of $\{f = a\}$ (see Section A of the Supplementary Material for a formal definition). We are interested in algorithms with rate-optimal worst-case sample complexity over classical function classes, as well as (slightly) improved sample complexity bounds in more favorable cases.

Main Contributions and Outline of the Paper.

- We define a new learning goal for level set approximation (see above and Section A of the Supplementary Material) similar in spirit to that of Gotovos et al. (2013).

- In Section 2 we briefly discuss the inherent hardness of the level set approximation problem (Theorem 1) and the role played by smoothness or structural assumptions on f .
- In Section 3 we design a family of algorithms called *Bisect and Approximate* through which we reduce the level set approximation problem to a local function approximation problem.
- In Sections 4 and 5 we instantiate Bisect and Approximate to the cases of Hölder or gradient-Hölder functions. We derive upper and lower bounds showing that the sample complexity for level set approximation is of the order of $1/\varepsilon^{d/\beta}$ in the worst-case, where $\beta \in (0, 2]$ is a smoothness parameter.
- In Section 5.2 we also show that Bisect and Approximate algorithms adapt to more favorable functions f by featuring a slightly improved sample complexity in such cases.

Some lemmas and proofs are deferred to the Supplementary Material.

Related Works. Sequential learning (sometimes referred to as sequential design of experiments) for level set and sublevel set identification is an active field of research. Many algorithms are based on Gaussian process priors over the black box function f (Ranjan et al., 2008; Vazquez and Bect, 2009; Picheny et al., 2010; Bect et al., 2012; Chevalier et al., 2014; Ginsbourger et al., 2014; Wang et al., 2016; Bect et al., 2017; Gotovos et al., 2013). In contrast with this large number of algorithms, few theoretical guarantees exist on the consistency or rate of convergence. Moreover, the majority of these guarantees are probabilistic. This means that consistency results state that an error goes to zero almost surely with respect to the Gaussian process prior measure over the unknown function f , and that the rates of convergence hold in probability, with respect to the same prior measure. In this probabilistic setting, Bect et al. (2019) provide a consistency result for a class of methods called Stepwise Uncertainty Reduction. Gotovos et al. (2013) provide rates of convergence, with noisy observations and for a classification-based loss function.

The loss function of Gotovos et al. (2013), given for sublevel set estimation, is similar in spirit to the notion of ε -approximation studied here for level set approximation, since we both aim at making decisions that are approximately correct for all x in the input space. The main difference is that Gotovos et al. (2013) assume that f is a realization of a Gaussian process and thus provide guarantees that are prob-

abilistic, while we prove deterministic bounds (for a fixed function). On the other hand, they consider noisy observations, while we assume f can be evaluated perfectly.

A related problem studied in statistics is density level set estimation, in which the superlevel set of a density f is estimated by looking at i.i.d. draws of random variables with density f . For this problem, several different performance measures are considered, such as the Hausdorff distance (Cadre et al., 2013; Singh et al., 2009; Tsybakov, 1997) or a measure of the symmetric difference (Cadre, 2006; Rigollet and Vert, 2009; Tsybakov, 1997).

When the function f is convex, our problem is also related to that of approximating a convex compact body with a simpler set (e.g., a polytope) in Hausdorff distance. This has been studied extensively in convex geometry and several sequential and non-sequential algorithms have been proposed (see, e.g., the two surveys Kamenev 2019; Gruber 1993 and references therein).

The closest connections with our work are within the bandit optimization literature. More precisely, our Bisect and Approximate algorithm and its analysis are inspired from the branch-and-bound algorithm of Locatelli and Carpentier (2018, Appendix A.2) and from the earlier methods of Perevozchikov (1990), Bubeck et al. (2011, HOO algorithm), and Munos et al. (2014, DOO algorithm). All these algorithms address the problem of finding a global extremum of f , while we are interested in finding level sets. However the idea of using a 2^d -ary tree to construct refined partitions of the input domain, and sequential methods to select which branch to explore next, are key in this paper.

There are also algorithmic connections with the non-parametric statistics literature. In particular, the idea of locally approximating a target function has been used many times for different purposes (e.g., Györfi et al. 2002; Tsybakov 2009).

Additional Notation. We denote the set of all positive integers $\{1, 2, \dots\}$ by \mathbb{N}^* . For all $x \in \mathbb{R}$, we denote by $\lceil x \rceil$ (resp., $\lfloor x \rfloor$) the ceiling (resp., floor) function at x , i.e., the smallest (resp., largest) integer larger (resp., smaller) or equal to x . Finally, for two sets A and B , we write $A \subseteq B$ to say that A is included in B (possibly with equality).

2 INHERENT HARDNESS

In this section we show that level sets are typically $(d - 1)$ -dimensional, and discuss the consequences of this fact in terms of the inherent hardness of the level set approximation problem.

We evaluate the dimension through the growth rate of packing numbers, one of the classical ways to measure the size of a set. In the case of the unit hypercube and the sup-norm, recall that packing numbers are defined as follows.

Definition 2 (Packing number). For all $r > 0$, the r -packing number $\mathcal{N}(E, r)$ of a subset E of $[0, 1]^d$ (with respect to the sup-norm $\|\cdot\|_\infty$) is the largest number of r -separated points contained in E , i.e.,

$$\mathcal{N}(E, r) := \sup\{k \in \mathbb{N}^* : \exists \mathbf{x}_1, \dots, \mathbf{x}_k \in E, \min_{i \neq j} \|\mathbf{x}_i - \mathbf{x}_j\|_\infty > r\} \quad (2)$$

if E is nonempty, zero otherwise.

The next theorem indicates that, with the exceptions of sets of minimizers or maximizers, ε -packing numbers of level sets $\{f = a\}$ of continuous functions f are at least $(d - 1)$ -dimensional. This result is very natural since $\{f = a\}$ is the solution set of one equation with d unknowns.

Theorem 1. Let $f: [0, 1]^d \rightarrow \mathbb{R}$ be a non-constant continuous function, and $a \in \mathbb{R}$ be any level such that $\min_{\mathbf{x} \in [0, 1]^d} f(\mathbf{x}) < a < \max_{\mathbf{x} \in [0, 1]^d} f(\mathbf{x})$. Then, there exists $\kappa > 0$ such that, for all $\varepsilon > 0$,

$$\mathcal{N}(\{f = a\}, \varepsilon) \geq \kappa \frac{1}{\varepsilon^{d-1}}.$$

We restate and prove this result in the Supplementary Material (Theorem 6, Section F.1).

We note an important difference with the global optimization problem. Indeed, the set of global maximizers (or minimizers) of a function f is typically finite and thus 0-dimensional. This implies that, depending on the shape of f around a global optimum, global optimization algorithms feature a sample complexity ranging roughly between $\log(1/\varepsilon)$ and $(1/\varepsilon)^d$ (see, e.g., Perevozchikov 1990; Munos et al. 2014).

In our case, by Theorem 1, level sets are large, so that we can expect the sample complexity to depend heavily on the input dimension d . This is however not the end of the story. Indeed, as in nonparametric statistics (e.g., Györfi et al. 2002; Tsybakov 2009) or in convex optimization (e.g., Nesterov 2004; Boyd

and Vandenberghe 2004; Bubeck 2015), additional smoothness or structural assumptions like convexity of f play a role in the hardness of the level set approximation problem. Since this problem is important in practice, designing algorithms that best exploit such additional assumptions is an important question. This is what we address in this paper.

3 BA ALGORITHMS & ANALYSIS

In this section, we introduce and analyze a family of algorithms designed for the problem of approximating the level set of an unknown function. They are based on an iterative refinement of the domain $[0, 1]^d$, as made precise in the following definition.

Definition 3 (Bisection of a family of hypercubes). Let \mathcal{C} be a family of n hypercubes included in $[0, 1]^d$. We say that $\text{bisect}(\mathcal{C})$ is the *bisection* of \mathcal{C} if it contains exactly the $2^d n$ hypercubes obtained by subdividing each $C = [a_1, b_1] \times \dots \times [a_d, b_d] \in \mathcal{C}$ into the 2^d equally-sized smaller hypercubes of the form $C' = I_1 \times \dots \times I_d$ with I_j being either $[a_j, (a_j + b_j)/2]$ or $[(a_j + b_j)/2, b_j]$.

Our algorithm is of the branch-and-bound type, similarly to other bandit algorithms for global optimization such as that of Locatelli and Carpentier (2018, Appendix A.2) and earlier methods (Perevozchikov, 1990; Bubeck et al., 2011; Munos et al., 2014).

Our Bisect and Approximate algorithms² (BA, Algorithm 2) maintain, at all iterations i , a collection \mathcal{C}_i of hypercubes on which the target function f is determined to take values close to the target level a . A BA algorithm takes as input the level a , a common number of queries k (to be performed in each hypercube at all iterations), and a pair of tolerance parameters $b, \beta > 0$, related to the smoothness of f and the approximation power of the approximators used by the algorithm. At the beginning of each iteration i , the collection of hypercubes \mathcal{C}_{i-1} determined at the end of the last iteration is bisected (line 3), so that all new hypercubes have diameter 2^{-i} (in the sup-norm). Then, the values of the target function f at k points of each newly created hypercube

²We refer to Bisect and Approximate algorithms in the plural form because different BA algorithms can be defined with the same input, depending on which rules are used to pick points at line 7 and approximators at line 10. E.g., BAH (Section 4) only looks at the center of each hypercube and uses constant approximators, while BAG (Section 5) queries the value of f at all vertices of each hypercube and builds higher-order polynomial approximators.

are queried (lines 7–8). The output set S_n after n queries to f is a subset of the union of all hypercubes in \mathcal{C}_{i-1} , i.e., the collection of all hypercubes determined during to the latest *completed* iteration. The precise definition of S_n depends on the approximators used during the last completed iteration and the two tolerance parameters b, β (lines 9 and 2).³ After all k values of f are queried from a hypercube C' , this information is used to determine a local approximator $g_{C'}$ of f (line 10). Finally, the collection of hypercubes \mathcal{C}_i is updated using $g_{C'}$ as a proxy for f (line 11) for all hypercubes C' . In this step, all hypercubes C' in which the proxy $g_{C'}$ is too far from the target level a are discarded, where the tightness of the rejection rule increases with the passing of the iterations i and it is further regulated by the two tolerance parameters b, β .

Algorithm 2: Bisect and Approximate (BA)

input: level $a \in \mathbb{R}$, queries $k \in \mathbb{N}^*$, tol. $b, \beta > 0$

init: $D \leftarrow [0, 1]^d$, $\mathcal{C}_0 \leftarrow \{D\}$, $g_D \equiv a$, $n \leftarrow 0$

```

1 for iteration  $i = 1, 2, \dots$  do
2    $S(i) \leftarrow \bigcup_{C' \in \mathcal{C}_{i-1}} \{\mathbf{x} \in C' : |g_{C'}(\mathbf{x}) - a| \leq b 2^{-\beta(i-1)}\}$ 
3    $\mathcal{C}'_i \leftarrow \text{bisect}(\mathcal{C}_{i-1})$ 
4   for each hypercube  $C' \in \mathcal{C}'_i$  do
5     for  $j = 1, \dots, k$  do
6       update  $n \leftarrow n + 1$ 
7       pick a query point  $\mathbf{x}_n \in C'$ 
8       observe  $f(\mathbf{x}_n)$ 
9       output  $S_n \leftarrow S(i)$ 
10    pick a local approximator  $g_{C'} : C' \rightarrow \mathbb{R}$ 
11     $\mathcal{C}_i \leftarrow \{C' \in \mathcal{C}'_i : \exists \mathbf{x} \in C', |g_{C'}(\mathbf{x}) - a| \leq b 2^{-\beta i}\}$ 

```

Our analysis of BA algorithms (Theorem 2) hinges on the accuracy of the approximators $g_{C'}$ selected at line 10, as formalized in the following definition.

Definition 4 (Accurate approximation). Let $b, \beta > 0$ and $C \subseteq [0, 1]^d$. We say that a function $g : C \rightarrow \mathbb{R}$ is a (b, β) -*accurate approximation* of another function $f : [0, 1]^d \rightarrow \mathbb{R}$ (on C) if the distance (in the sup-norm on C) between f and g can be controlled with the diameter (in the sup-norm) of C as

$$\sup_{\mathbf{x} \in C} |g(\mathbf{x}) - f(\mathbf{x})| \leq b \left(\sup_{\mathbf{x}, \mathbf{y} \in C} \|\mathbf{x} - \mathbf{y}\|_\infty \right)^\beta.$$

We now present one of our main results, which states that BA algorithms run with accurate approximations of the target function return ε -approximations

³Notably, the output set $S(i)$ at iteration i can be represented succinctly and testing if $\mathbf{x} \in S(i)$ can be done efficiently in all our BA instances in Sections 4, 5.

of the target level set after a number of queries that depends on the packing number (Definition 2) of the inflated level set at decreasing scales.⁴

Theorem 2. *Consider a Bisect and Approximate algorithm (Algorithm 2) run with input a, k, b, β . Let $f: [0, 1]^d \rightarrow \mathbb{R}$ be an arbitrary function with level set $\{f = a\} \neq \emptyset$. Assume that the approximators $g_{C'}$ selected at line 10 are (b, β) -accurate approximations of f (Definition 4). Fix any accuracy $\varepsilon > 0$, let $i(\varepsilon) := \lceil (1/\beta) \log_2(2b/\varepsilon) \rceil$, and define $n(\varepsilon)$ by⁵*

$$4^d k \sum_{i=0}^{i(\varepsilon)-1} \lim_{\delta \rightarrow 1^-} \mathcal{N}(\{|f - a| \leq 2b2^{-\beta i}\}, \delta 2^{-i}). \quad (3)$$

Then, for all $n > n(\varepsilon)$, the output S_n returned after the n -th query is an ε -approximation of $\{f = a\}$.

The expression (3) can be simplified by taking $\delta = 1$ and increasing the leading multiplicative constant. However, in the following sections we will see how to upper bound this quantity with simpler functions of $1/\varepsilon$, for which the limit can be computed exactly.

Proof. Fix any $n > n(\varepsilon)$. We begin by proving that

$$\{f = a\} \subseteq S_n. \quad (4)$$

Recall that $S_n = \bigcup_{C \in \mathcal{C}_{i-1}} \{\mathbf{x} \in C : |g_C(\mathbf{x}) - a| \leq b2^{-\beta(\iota-1)}\}$ (line 9), where $\iota = \iota(n)$ is the iteration during which the n -th value of f is queried. To prove (4), we will show the stronger result: for all $i \geq 0$,

$$\{f = a\} \subseteq \bigcup_{C \in \mathcal{C}_i} \{\mathbf{x} \in C : |g_C(\mathbf{x}) - a| \leq b2^{-\beta i}\}, \quad (5)$$

i.e., that the level set $\{f = a\}$ is *always* included in the output set, not only after iteration $\iota(n) - 1$ has been completed. We do so by induction. If $i = 0$, then $\{f = a\} \subseteq [0, 1]^d = \{\mathbf{x} \in [0, 1]^d : |a - a| \leq b2^{-\beta \cdot 0}\}$, which is the union in (5) by definition of $D = [0, 1]^d$, $\mathcal{C}_0 = \{D\}$ and $g_D \equiv a$ in the initialization of Algorithm 2. Assume now that the inclusion holds for some $i - 1$: we will show that it keeps holding for the next iteration $i \in \mathbb{N}^*$. Indeed, fix any $\mathbf{z} \in \{f = a\}$. By induction, \mathbf{z} belongs to some hypercube $C \in \mathcal{C}_{i-1}$. Since $\mathcal{C}'_i = \text{bisect}(\mathcal{C}_{i-1})$ (line 3), by definition of bisection (Definition 3) we

⁴Note that our BA algorithm can be used for estimating a sublevel set $\{f \leq a\}$ by simply dropping the absolute values in lines 2 and 11 of Algorithm 2. As the reader might realize, the same proof techniques would apply with the corresponding (straightforward) changes. An analogous argument applies to superlevel sets $\{f \geq a\}$.

⁵Letting $\sum_{i=0}^m a_i = 0$ for any $m > 0$ and all $a_j \in \mathbb{R}$.

have that $\bigcup_{C' \in \mathcal{C}'_i} C' = \bigcup_{C \in \mathcal{C}_{i-1}} C$, which in turns implies that there exists a hypercube $C'_z \in \mathcal{C}'_i$ such that $\mathbf{z} \in C'_z$. We show now that this C'_z also belongs to \mathcal{C}_i , i.e., that it is not discarded during the update of the algorithm at line 11. Indeed, since $g_{C'_z}$ is a (b, β) -accurate approximation of f on C'_z (by assumption) and the diameter (in the sup-norm) of C'_z is $\sup_{\mathbf{x}, \mathbf{y} \in C'_z} \|\mathbf{x} - \mathbf{y}\|_\infty = 2^{-i}$, we have that $|g_{C'_z}(\mathbf{z}) - a| = |g_{C'_z}(\mathbf{z}) - f(\mathbf{z})| \leq b2^{-\beta i}$. This gives both that $\mathbf{z} \in C'_z \in \mathcal{C}_i$ (by definition of \mathcal{C}_i at line 11) and, consequently, that $\mathbf{z} \in \bigcup_{C \in \mathcal{C}_i} \{\mathbf{x} \in C : |g_C(\mathbf{x}) - a| \leq b2^{-\beta i}\}$, which clinches the proof of (5) and in turn yields (4).

We now show the validity of the second inclusion

$$S_n \subseteq \{|f - a| \leq \varepsilon\}. \quad (6)$$

As above, let $\iota = \iota(n)$ be the iteration during which the n -th value of f is queried by the algorithm. Fix any $\mathbf{z} \in S_n$. We will prove that $\mathbf{z} \in \{|f - a| \leq \varepsilon\}$ or, restated equivalently, that $|f(\mathbf{z}) - a| \leq \varepsilon$. By definition of $S_n = \bigcup_{C \in \mathcal{C}_{i-1}} \{\mathbf{x} \in C : |g_C(\mathbf{x}) - a| \leq b2^{-\beta(\iota-1)}\}$ (line 9), since $\mathbf{z} \in S_n$, then there exists $C_z \in \mathcal{C}_{i-1}$ such that $\mathbf{z} \in C_z$ and $|g_{C_z}(\mathbf{z}) - a| \leq b2^{-\beta(\iota-1)}$. Moreover, since $C_z \in \mathcal{C}_{i-1} \subseteq \mathcal{C}'_{i-1}$ has diameter $\sup_{\mathbf{x}, \mathbf{y} \in C_z} \|\mathbf{x} - \mathbf{y}\|_\infty = 2^{-(\iota-1)}$ (in the sup-norm) and the approximator g_{C_z} is a (b, β) -accurate approximation of f on C_z (by assumption), we have that $|f(\mathbf{z}) - g_{C_z}(\mathbf{z})| \leq b2^{-\beta(\iota-1)}$. Thus

$$|f(\mathbf{z}) - a| \leq |f(\mathbf{z}) - g_{C_z}(\mathbf{z})| + |g_{C_z}(\mathbf{z}) - a| \leq 2b2^{-\beta(\iota-1)}$$

and the right-hand side would be smaller than ε — proving (6) — if either $\varepsilon \geq 2b$ (trivially), or in case $\varepsilon \in (0, 2b)$, if we could guarantee that the iteration $\iota = \iota(n)$ during which the n -th value of f is queried satisfies $\iota - 1 \geq \lceil (1/\beta) \log_2(2b/\varepsilon) \rceil = i(\varepsilon)$. In other words, assuming without loss of generality that $\varepsilon \in (0, 2b)$ (so that $i(\varepsilon) \geq 1$) and recalling that $n > n(\varepsilon)$, in order to prove (6) we only need to check that the $i(\varepsilon)$ -th iteration is guaranteed to be concluded after at most $n(\varepsilon)$ queries, where $n(\varepsilon)$ is defined in terms of packing numbers in (3). To see this, note that the total number of values of f that the algorithm queries by the end of iteration $i(\varepsilon)$ is $\sum_{i=1}^{i(\varepsilon)} k |\mathcal{C}'_i| = 2^d k \sum_{i=1}^{i(\varepsilon)} |\mathcal{C}_{i-1}| = 2^d k \sum_{i=0}^{i(\varepsilon)-1} |\mathcal{C}_i|$. To conclude the proof, it is now sufficient to show that for all iterations $i \geq 0$, the number of hypercubes maintained by the algorithm can be upper bounded by

$$|\mathcal{C}_i| \leq 2^d \lim_{\delta \rightarrow 1^-} \mathcal{N}(\{|f - a| \leq 2b2^{-\beta i}\}, \delta 2^{-i}). \quad (7)$$

Fix an arbitrary $\delta \in (0, 1)$. If $i = 0$, then $|\mathcal{C}_0| = 1 \leq \mathcal{N}(\{|f - a| \leq 2b\}, \delta)$ by the definitions of

$\mathcal{C}_0 = \{[0, 1]^d\}$ (initialization of Algorithm 2) and δ -packing number (Definition 2) of $\{|f - a| \leq 2b\}$ (which is non-empty because it contains $\{f = a\}$). Fix any iteration $i \in \mathbb{N}^*$. By definition of \mathcal{C}_i (line 11), for all hypercubes $C \in \mathcal{C}_i$ there exists a point $\mathbf{x}_C \in C$ such that $|g_C(\mathbf{x}_C) - a| \leq b2^{-\beta i}$. Hence, for each hypercube $C \in \mathcal{C}_i$ there exists one of its points $\mathbf{x}_C \in C$ such that $|f(\mathbf{x}_C) - a|$ can be upper bounded by

$$|f(\mathbf{x}_C) - g_C(\mathbf{x}_C)| + |g_C(\mathbf{x}_C) - a| \leq 2b2^{-\beta i}, \quad (8)$$

where, recalling that all hypercubes in \mathcal{C}_i have diameter 2^{-i} (in the sup-norm), the bound on the term $|f(\mathbf{x}_C) - g_C(\mathbf{x}_C)|$ is a consequence of g_C being a (b, β) -accurate approximation of f on C .

Now we claim that the family of hypercubes \mathcal{C}_i can be partitioned into 2^d subfamilies $\mathcal{C}_i(1), \dots, \mathcal{C}_i(2^d)$ with the property that all distinct hypercubes $C \neq C'$ belonging to the same family $\mathcal{C}_i(k)$ are strictly $(\delta 2^{-i})$ -separated (in the sup-norm), i.e., that for all $k \in \{1, \dots, 2^d\}$ and all $C, C' \in \mathcal{C}_i(k)$, $C \neq C'$, we have $\inf_{\mathbf{x} \in C, \mathbf{y} \in C'} \|\mathbf{x} - \mathbf{y}\|_\infty > \delta 2^{-i}$.

We defer the proof of this claim to Section C of the Supplementary Material (for an insightful picture, see Figure 1 in the same section). Assume for now that it is true and fix an arbitrary $k \in \{1, \dots, 2^d\}$. Then, for all $C \in \mathcal{C}_i(k)$, there exists \mathbf{x}_C such that (8) holds. Therefore, we determined the existence of $|\mathcal{C}_i(k)|$ -many $(\delta 2^{-i})$ -separated points that are all included in $\{|f - a| \leq 2b2^{-\beta i}\}$. By definition of $(\delta 2^{-i})$ -packing number of $\{|f - a| \leq 2b2^{-\beta i}\}$ (i.e., the *largest* cardinality of a set of $(\delta 2^{-i})$ -separated points included in $\{|f - a| \leq 2b2^{-\beta i}\}$ — Definition 2), this implies that $|\mathcal{C}_i(k)| \leq \mathcal{N}(\{|f - a| \leq 2b2^{-\beta i}\}, \delta 2^{-i})$. Recalling that $\mathcal{C}_i(1), \dots, \mathcal{C}_i(2^d)$ is a partition of \mathcal{C}_i , we then obtain

$$|\mathcal{C}_i| = \sum_{k=1}^{2^d} |\mathcal{C}_i(k)| \leq 2^d \mathcal{N}(\{|f - a| \leq 2b2^{-\beta i}\}, \delta 2^{-i})$$

which, after taking the infimum over $\delta \in (0, 1)$ and by the monotonicity of the packing number $r \mapsto \mathcal{N}(E, r)$ (for any $E \subseteq [0, 1]^d$), yields

$$\begin{aligned} |\mathcal{C}_i| &\leq \inf_{\delta \in (0, 1)} \left(2^d \mathcal{N}(\{|f - a| \leq 2b2^{-\beta i}\}, \delta 2^{-i}) \right) \\ &= 2^d \lim_{\delta \rightarrow 1^-} \mathcal{N}(\{|f - a| \leq 2b2^{-\beta i}\}, \delta 2^{-i}). \end{aligned}$$

This gives (7) and concludes the proof. \square

By looking at the end of the proof of the previous result, one could see that the exponential term 4^d in

our bound (3) could be lowered to 2^d under the assumption that at any iteration i , Algorithm 2 picks at least one $\mathbf{x}_{C'} \in C'$ for each $C' \in \mathcal{C}'_i$ such that $\|\mathbf{x}_{C'_1} - \mathbf{x}_{C'_2}\|_\infty \geq 2^{-i}$ for all distinct $C'_1, C'_2 \in \mathcal{C}'_i$. Notably this property is enjoyed by all our BA instances in Sections 4, 5.

4 HÖLDER FUNCTIONS

In this section, we focus on Hölder functions, and we present a BA instance that is rate-optimal for determining their level sets.

Definition 5 (Hölder function). Let $c > 0$, $\gamma \in (0, 1]$, and $E \subseteq [0, 1]^d$. We say that a function $f: E \rightarrow \mathbb{R}$ is (c, γ) -Hölder (with respect to the sup-norm $\|\cdot\|_\infty$) if $|f(\mathbf{x}) - f(\mathbf{y})| \leq c \|\mathbf{x} - \mathbf{y}\|_\infty^\gamma$, for all $\mathbf{x}, \mathbf{y} \in E$.

Our BA instance for Hölder functions (BAH, Algorithm 3) runs Algorithm 2 with $k = 1$, $b = c$, $\beta = \gamma$. The local approximators $g_{C'}$ are constant and equal to the value $f(\mathbf{c}_{C'})$ at the center $\mathbf{c}_{C'}$ of C' . In particular, the output set S_n is now the entire union of all hypercubes determined in the latest completed iteration.

Algorithm 3: BA for Hölder Functions (BAH)

input: level $a \in \mathbb{R}$, tol. $c > 0$, $\gamma \in (0, 1]$
init: $D \leftarrow [0, 1]^d$, $\mathcal{C}_0 \leftarrow \{D\}$, $n \leftarrow 0$
1 for iteration $i = 1, 2, \dots$ **do**
2 $S(i) \leftarrow \bigcup_{C \in \mathcal{C}_{i-1}} C$
3 let $\mathcal{C}'_i \leftarrow \text{bisect}(\mathcal{C}_{i-1})$
4 **for each hypercube** $C' \in \mathcal{C}'_i$ **do**
5 update $n \leftarrow n + 1$
6 pick the center $\mathbf{c}_{C'}$ of C' as the next \mathbf{x}_n
7 observe $f(\mathbf{x}_n)$
8 output $S_n \leftarrow S(i)$
9 $\mathcal{C}_i \leftarrow \{C' \in \mathcal{C}'_i : |f(\mathbf{c}_{C'}) - a| \leq c2^{-\gamma i}\}$

The next result shows that the optimal worst-case sample complexity of the level set approximation of Hölder functions is of order $1/\varepsilon^{d/\gamma}$, and it is attained by BAH (Algorithm 3).

Theorem 3. *Let $a \in \mathbb{R}$, $c > 0$, $\gamma \in (0, 1]$, and $f: [0, 1]^d \rightarrow \mathbb{R}$ be any (c, γ) -Hölder function with level set $\{f = a\} \neq \emptyset$. Fix any accuracy $\varepsilon > 0$. Then, there exists $\kappa_1 > 0$ (independent of ε) such that, for all $n > \kappa_1/\varepsilon^{d/\gamma}$, the output S_n returned by BAH after the n -th query is an ε -approximation of $\{f = a\}$.*

Moreover, there exists κ_2 (independent of ε) such that no deterministic algorithm can guarantee to out-

put an ε -approximation of the level set $\{f = a\}$ for all (c, γ) -Hölder functions f , querying less than $\kappa_2/\varepsilon^{d/\gamma}$ of their values.

The proof is deferred to Section D in the Supplementary Material. The upper bound is an application of Theorem 2. The lower bound is proven by showing that no algorithm can distinguish between the function $f \equiv 0$ and a function that is non-zero only on a small ball, on which it attains the value 2ε . We use a classical construction with bump functions that appears, e.g., in Theorem 3.2 of Györfi et al. (2002) for nonparametric regression lower bounds.

We remark that the rate in the previous result is the same as that of a naive uniform grid filling of the space with step-size of order $\varepsilon^{1/\gamma}$. While this rate cannot be improved in the worst case, the leading constant of our sequential algorithm may be better if a large fraction of the input space can be rejected quickly. More importantly, we will see in Section 5.2 that (slightly) better rates can be attained by BA algorithms if the inflated level sets of the target function f are smaller (as it happens, e.g., if f is convex with proper level set $\{f = a\}$).

In the following section, we will also investigate if and to what extent higher smoothness helps. To this end, we will switch our focus to differentiable functions with Hölder gradients.

5 ∇ -HÖLDER FUNCTIONS: BAG ALGORITHM & ANALYSIS

In this section, we focus on differentiable functions with Hölder gradients, and we present a BA instance that is rate-optimal for determining their level sets.

Definition 6 (Gradient-Hölder/Lipschitz function). Let $c_1 > 0$, $\gamma_1 \in (0, 1]$, and $E \subseteq [0, 1]^d$. We say that a function $f: E \rightarrow \mathbb{R}$ is (c_1, γ_1) -gradient-Hölder (with respect to $\|\cdot\|_\infty$) if it is the restriction⁶ (to E) of a continuously differentiable function defined on \mathbb{R}^d such that $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_\infty \leq c_1 \|\mathbf{x} - \mathbf{y}\|_\infty^{\gamma_1}$ for all $\mathbf{x}, \mathbf{y} \in E$. If $\gamma_1 = 1$, we say that f is c_1 -gradient-Lipschitz.

The next lemma introduces the polynomial approximators that will be used by our BA instance and it shows that they are $(c_1 d, 1 + \gamma_1)$ -accurate approximations of f on all hypercubes.

Lemma 1 (BAG approximators). *Let $f: C' \rightarrow \mathbb{R}$ be a (c_1, γ_1) -gradient-Hölder function, for some $c_1 >$*

⁶They are defined as restrictions of continuously differentiable functions in order to have simply and well-defined gradients on the boundary of their domains.

0 and $\gamma_1 \in (0, 1]$. Let $C' \subseteq [0, 1]^d$ be a hypercube with diameter $\ell \in (0, 1]$ and set of vertices V' , i.e., $C' = \prod_{j=1}^d [u_j, u_j + \ell]$, for some $\mathbf{u} := (u_1, \dots, u_d) \in [0, 1 - \ell]^d$, and $V' = \prod_{j=1}^d \{u_j, u_j + \ell\}$. The function

$$h_{C'}: C' \rightarrow \mathbb{R} \\ \mathbf{x} \mapsto \sum_{\mathbf{v} \in V'} f(\mathbf{v}) \prod_{j=1}^d p_{v_j}(x_j), \quad (9)$$

where

$$p_{v_j}(x_j) := \left(1 - \frac{x_j - u_j}{\ell}\right) \mathbb{I}_{v_j = u_j} + \frac{x_j - u_j}{\ell} \mathbb{I}_{v_j = u_j + \ell},$$

interpolates the 2^d pairs $\{(\mathbf{v}, f(\mathbf{v}))\}_{\mathbf{v} \in V'}$ and it satisfies

$$\sup_{\mathbf{x} \in C'} |h_{C'}(\mathbf{x}) - f(\mathbf{x})| \leq c_1 d \ell^{1+\gamma_1}.$$

The technical proof of the previous lemma is deferred to Section E of the Supplementary Material.

Our Bisect and Approximate instance for gradient-Hölder functions (BAG) runs Algorithm 2 with $k = 2^d$, $b = c_1 d$, and $\beta = 1 + \gamma_1$. The local approximators $h_{C'}$ (defined in (9)) are computed by querying the values of f at all vertices of C' . Note that line 12 of Algorithm 4 can be carried out efficiently since it is sufficient to check the condition on $|h_{C'}(\mathbf{x}) - a|$ at the vertices \mathbf{x} of C' .⁷ Also, note that the output set S_n is the union over hypercubes of pre-images of segments from the polynomial functions in (9).

5.1 Worst-Case Sample Complexity

The next result shows that the optimal worst-case sample complexity of the level set approximation of gradient-Hölder functions is of order $1/\varepsilon^{d/(1+\gamma_1)}$, and it is attained by BAG (Algorithm 4).

Theorem 4. *Let $a \in \mathbb{R}$, $c_1 > 0$, $\gamma_1 \in (0, 1]$, and $f: [0, 1]^d \rightarrow \mathbb{R}$ be any (c_1, γ_1) -gradient-Hölder function with level set $\{f = a\} \neq \emptyset$. Fix any accuracy $\varepsilon > 0$. Then, there exists $\kappa_1 > 0$ (independent of ε) such that, for all $n > \kappa_1/\varepsilon^{d/(1+\gamma_1)}$, the output*

⁷Indeed, only three cases can occur. We set $\rho = c_1 d 2^{-(1+\gamma_1)^i}$. Case 1: if one of the vertices \mathbf{x} satisfies $|h_{C'}(\mathbf{x}) - a| \leq \rho$, then the condition is checked. Case 2: if the values $h_{C'}(\mathbf{x})$ at the vertices are all strictly below $a - \rho$ or all strictly above $a + \rho$, then it is also the case for all $\mathbf{x} \in C'$, since $h_{C'}(\mathbf{x})$ is a convex combination of all values at the vertices; so the condition is not checked. Case 3: if there are two vertices \mathbf{x} and \mathbf{y} such that $h_{C'}(\mathbf{x}) < a - \rho$ and $h_{C'}(\mathbf{y}) > a + \rho$, then there exists $\mathbf{z} \in C'$ such that $|h_{C'}(\mathbf{z}) - a| \leq \rho$ by continuity of $h_{C'}$ on C' ; so the condition is checked.

Algorithm 4: BA for ∇ -Hölder f (BAG)

input: level $a \in \mathbb{R}$, tol. $c_1 > 0$, $\gamma_1 \in (0, 1]$
init: $D \leftarrow [0, 1]^d$, $\mathcal{C}_0 \leftarrow \{D\}$, $h_D \equiv a$, $n \leftarrow 0$
1 **for** iteration $i = 1, 2, \dots$ **do**
2 $S(i) \leftarrow \bigcup_{C \in \mathcal{C}_{i-1}} \{\mathbf{x} \in C : |h_C(\mathbf{x}) - a| \leq$
 $c_1 d 2^{-(1+\gamma_1)(i-1)}\}$
3 $\mathcal{C}'_i \leftarrow \text{bisect}(\mathcal{C}_{i-1})$
4 **for each** hypercube $C' \in \mathcal{C}'_i$ **do**
5 let $V' \subseteq C'$ be the set of vertices of C'
6 **for each** vertex $\mathbf{v} \in V'$ **do**
7 update $n \leftarrow n + 1$
8 pick vertex $\mathbf{v} \in V'$ as the next \mathbf{x}_n
9 observe $f(\mathbf{x}_n)$
10 output $S_n \leftarrow S(i)$
11 interpolate the 2^d pairs $\{(\mathbf{v}, f(\mathbf{v}))\}_{\mathbf{v} \in V'}$
 with $h_{C'} : C' \rightarrow \mathbb{R}$ given by (9)
12 update $\mathcal{C}_i \leftarrow \{C' \in \mathcal{C}'_i : \text{there exists } \mathbf{x} \in$
 $C' \text{ such that } |h_{C'}(\mathbf{x}) - a| \leq c_1 d 2^{-(1+\gamma_1)i}\}$

S_n returned by BAG after the n -th query is an ε -approximation of $\{f = a\}$.

Moreover, there exists $\kappa_2 > 0$ (independent of ε) such that no deterministic algorithm can guarantee to output an ε -approximation of the level set $\{f = a\}$ for all (c_1, γ_1) -gradient-Hölder functions f , querying less than $\kappa_2/\varepsilon^{d/(1+\gamma_1)}$ of their values.

The proof proceeds similarly to that of Theorem 3. It is deferred to Section E of the Supplementary Material.

Similarly to Section 4, the rate in the previous result could also be achieved by choosing query points on a regular grid with step-size of order $\varepsilon^{1/(1+\gamma_1)}$. However, our sequential algorithm features an improved sample complexity outside of a worst-case scenario, as shown in the following section.

5.2 Adaptivity to Smaller d^*

Our general result (Theorem 2) suggests that the sample complexity can be controlled whenever there exists $d^* \geq 0$ such that

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

for some $C^* > 0$. We call such a d^* a *NLS dimension* of $\{f = a\}$. Note that such a d^* always exists and $d^* \leq d$ by $\{|f - a| \leq r\} \subseteq [0, 1]^d$. However $d^* \geq d - 1$ by Theorem 1 for non-degenerate level sets of continuous functions (for more details, see Section F.1 in the Supplementary Material). The

definition of NLS dimension leads to the following result.

Corollary 1. *Let $a \in \mathbb{R}, c_1 > 0, \gamma_1 \in (0, 1]$, and $f : [0, 1]^d \rightarrow \mathbb{R}$ be any (c_1, γ_1) -gradient-Hölder function with level set $\{f = a\} \neq \emptyset$. Let $d^* \in [d - 1, d]$ be a NLS dimension of $\{f = a\}$. Fix any accuracy $\varepsilon > 0$. Then, for all $n > m(\varepsilon)$, the output S_n returned by BAG after the n -th query is an ε -approximation of $\{f = a\}$, where*

$$m(\varepsilon) := \begin{cases} \kappa_1 + \kappa_2 \log_2 \left(\frac{1}{\varepsilon^{1/(1+\gamma_1)}} \right)^+ & \text{if } d^* = 0, \\ \kappa(d^*) \frac{1}{\varepsilon^{d^*/(1+\gamma_1)}} & \text{if } d^* > 0, \end{cases}$$

for $\kappa_1, \kappa_2, \kappa(d^*) \geq 0$ independent of ε , that depend exponentially on d , where $x^+ = \max\{x, 0\}$.

We remark that $d^* = d - 1$ can be achieved by well-behaved functions. This is typically the case when f is convex or (as a corollary) if it consists of finitely many convex components.⁸ This non-trivial claim is proved in Section F.2 of the Supplementary Material for convex functions with a proper level set⁹. The following result, combined with this fact and Corollary 1, shows that BAG is rate-optimal for determining proper level sets of convex gradient-Lipschitz functions.

Theorem 5. *Fix any level $a \in \mathbb{R}$ and an arbitrary accuracy $\varepsilon > 0$. No deterministic algorithm A can guarantee to output an ε -approximation of any Δ -proper level set $\{f = a\}$ of an arbitrary convex c_1 -gradient-Lipschitz functions f with $c_1 \geq 3$ and $\Delta \in (0, 1/4]$, querying less than $\kappa/\varepsilon^{(d-1)/2}$ of their values, where $\kappa > 0$ is a constant independent of ε .*

We give a complete proof of this result in Section F.3 of the Supplementary Material.

6 CONCLUSION

We studied the problem of determining ε -approximations of the level set of a target function f by only querying its values. After discussing the inherent hardness of the problem (Theorem 1), we designed the class of BA algorithms for which we proved theoretical guarantees under the assumption that accurate local approximations of f can be computed by only looking at its values (Theorem 2).

⁸More precisely, if for some $a' > a$, we have that the sublevel $\{f \leq a'\}$ is a disjoint union of a finite number of convex sets on which f is convex.

⁹ $\{f = a\}$ is Δ -proper for some $\Delta > 0$ if we have $\min_{\mathbf{x} \in [0, 1]^d} f(\mathbf{x}) + \Delta \leq a \leq \max_{\mathbf{x} \in \partial[0, 1]^d} f(\mathbf{x})$.

This provides a general method to reduce our level set approximation problem to a local approximation problem, decoupled from the original one.

Such an approach leads to rate-optimal worst-case sample complexity guarantees for the case of Hölder and gradient-Hölder functions (Theorems 3, 4). At the same time, we show that in some cases our BA algorithms adapt to a natural structural property of f , namely small NLS dimension (Corollary 1) including convexity (Theorem 5 and preceding discussion).

Future Work. Compared to the best achievable rate $1/\varepsilon^{d/\gamma}$ for (c, γ) -Hölder functions, we show that BA algorithms converge at a faster $1/\varepsilon^{d/(1+\gamma_1)}$ rate if f is (c_1, γ_1) -gradient-Hölder. This points at an interesting line of research: the study of general Hölder spaces in which the target function is k times continuously differentiable and the k -th partial derivatives are (c_k, γ_k) -Hölder, for some $k \in \mathbb{N}^*$, $c_k > 0$, and $\gamma_k \in (0, 1]$. We conjecture that a suitable choice of approximators for our BA algorithms would lead to a rate-optimal sample complexity of order $1/\varepsilon^{d/(k+\gamma_k)}$ for this class of functions, making optimal solutions for this problem sample-efficient. Another possible line of research is the design of algorithms that adapt to the smoothness of f when the latter is unknown, similarly to global bandit optimization (e.g., Grill et al. 2015; Bartlett et al. 2019). We leave these interesting directions open for future work.

Acknowledgments

The work of Tommaso Cesari and Sébastien Gerchinovitz has benefited from the AI Interdisciplinary Institute ANITI, which is funded by the French “Investing for the Future – PIA3” program under the Grant agreement ANR-19-P3IA-0004. Sébastien Gerchinovitz gratefully acknowledges the support of the DEEL project (<https://www.deel.ai/>). This work benefited from the support of the project BOLD from the French national research agency (ANR).

References

Dario Azzimonti, David Ginsbourger, Clément Chevalier, Julien Bect, and Yann Richet. Adaptive design of experiments for conservative estimation of excursion sets. *Technometrics*, pages 1–14, 2020.

Peter L. Bartlett, Victor Gabillon, and Michal Valko. A simple parameter-free and adaptive approach to optimization under a minimal local

smoothness assumption. In *Algorithmic Learning Theory*, pages 184–206, 2019.

Julien Bect, David Ginsbourger, Ling Li, Victor Picheny, and Emmanuel Vazquez. Sequential design of computer experiments for the estimation of a probability of failure. *Statistics and Computing*, 22(3):773–793, 2012.

Julien Bect, Ling Li, and Emmanuel Vazquez. Bayesian subset simulation. *SIAM/ASA Journal on Uncertainty Quantification*, 5(1):762–786, 2017.

Julien Bect, François Bachoc, David Ginsbourger, et al. A supermartingale approach to Gaussian process based sequential design of experiments. *Bernoulli*, 25(4A):2883–2919, 2019.

Clément Bouttier, Tommaso Cesari, and Sébastien Gerchinovitz. Regret analysis of the Piyavskii–Shubert algorithm for global Lipschitz optimization. *arXiv preprint arXiv:2002.02390*, 2020.

Stephen Boyd and Lieven Vandenbergh. *Convex optimization*. Cambridge University Press, 2004.

Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, 8(3-4):231–357, 2015. ISSN 1935-8237.

Sébastien Bubeck, Rémi Munos, Gilles Stoltz, and Csaba Szepesvári. X-armed bandits. *Journal of Machine Learning Research*, 12(May):1655–1695, 2011.

Benoît Cadre. Kernel estimation of density level sets. *Journal of multivariate analysis*, 97(4):999–1023, 2006.

Benoît Cadre, Bruno Pelletier, and Pierre Pudlo. Estimation of density level sets with a given probability content. *Journal of Nonparametric Statistics*, 25(1):261–272, 2013.

Clément Chevalier, Julien Bect, David Ginsbourger, Emmanuel Vazquez, Victor Picheny, and Yann Richet. Fast parallel kriging-based stepwise uncertainty reduction with application to the identification of an excursion set. *Technometrics*, 56(4):455–465, 2014.

Frédéric Galland, Philippe Réfrégier, and Olivier Germain. Synthetic aperture radar oil spill segmentation by stochastic complexity minimization. *IEEE Geoscience and Remote Sensing Letters*, 1(4):295–299, 2004.

David Ginsbourger, Jean Baccou, Clément Chevalier, Frédéric Perales, Nicolas Garland, and Yann Monerie. Bayesian adaptive reconstruction

- of profile optima and optimizers. *SIAM/ASA Journal on Uncertainty Quantification*, 2(1):490–510, 2014.
- Alkis Gotovos, Nathalie Casati, Gregory Hitz, and Andreas Krause. Active learning for level set estimation. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 1344–1350. AAAI Press, 2013.
- Jean-Bastien Grill, Michal Valko, and Rémi Munos. Black-box optimization of noisy functions with unknown smoothness. In *Advances in Neural Information Processing Systems*, pages 667–675, 2015.
- Peter Gruber. Aspects of approximation of convex bodies. In *Handbook of convex geometry*, pages 319–345. Elsevier, 1993.
- László Györfi, Adam Krzyżak, Michael Kohler, and Harro Walk. *A Distribution-Free Theory of Non-parametric Regression*. Springer Series in Statistics. Springer-Verlag, New York, 2002.
- George Kamenev. Optimal non-adaptive approximation of convex bodies by polytopes. In *Numerical Geometry, Grid Generation and Scientific Computing*, pages 157–172. Springer, 2019.
- Andrea Locatelli and Alexandra Carpentier. Adaptivity to smoothness in X-armed bandits. In Sébastien Bubeck, Vianney Perchet, and Philippe Rigollet, editors, *Conference on Learning Theory*, volume 75 of *Proceedings of Machine Learning Research*, pages 1463–1492. PMLR, 06–09 Jul 2018.
- Rémi Munos et al. From bandits to Monte-Carlo Tree search: The optimistic principle applied to optimization and planning. *Foundations and Trends® in Machine Learning*, 7(1):1–129, 2014.
- Yurii Nesterov. *Introductory lectures on convex optimization: a basic course*, volume 87 of *Applied Optimization*. Springer US, 2004.
- Alexander G. Perevozchikov. The complexity of the computation of the global extremum in a class of multi-extremum problems. *USSR Computational Mathematics and Mathematical Physics*, 30(2):28–33, 1990.
- Victor Picheny, David Ginsbourger, Olivier Roustant, Raphael Haftka, and Nam-Ho Kim. Adaptive designs of experiments for accurate approximation of a target region. *Journal of Mechanical Design*, 132(7), 2010.
- Remy Priem, Hugo Gagnon, Ian Chittick, Stephane Dufresne, Youssef Diouane, and Nathalie Bartoli. An efficient application of Bayesian optimization to an industrial MDO framework for aircraft design. In *AIAA AVIATION 2020 FORUM*, page 3152, 2020.
- Mohammad Rahimi, Richard Pon, William Kaiser, Gaurav Sukhatme, Deborah Estrin, and Mani Srivastava. Adaptive sampling for environmental robotics. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, volume 4, pages 3537–3544. IEEE, 2004.
- Pritam Ranjan, Derek Bingham, and George Michailidis. Sequential experiment design for contour estimation from complex computer codes. *Technometrics*, 50(4):527–541, 2008.
- Philippe Rigollet and Regis Vert. Optimal rates for plug-in estimators of density level sets. *Bernoulli*, 15(4):1154–1178, 2009.
- Jerome Sacks, William Welch, Toby Mitchell, and Henry Wynn. Design and analysis of computer experiments. *Statistical science*, pages 409–423, 1989.
- Thomas Santner, Brian Williams, William Notz, and Brain Williams. *The design and analysis of computer experiments*, volume 1. Springer, 2003.
- Aarti Singh, Clayton Scott, Robert Nowak, et al. Adaptive Hausdorff estimation of density level sets. *The Annals of Statistics*, 37(5B):2760–2782, 2009.
- Alexandre B. Tsybakov. On nonparametric estimation of density level sets. *The Annals of Statistics*, 25(3):948–969, 1997.
- Alexandre B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer, 2009.
- Emmanuel Vazquez and Julien Bect. A sequential Bayesian algorithm to estimate a probability of failure. *IFAC Proceedings Volumes*, 42(10):546–550, 2009.
- Martin Wainwright. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2019.
- Hongqiao Wang, Guang Lin, and Jinglai Li. Gaussian process surrogates for failure detection: A Bayesian experimental design approach. *Journal of Computational Physics*, 313:247–259, 2016.