



HAL
open science

Sequential design of experiments for estimating percentiles of black-box functions

Tatiana Labopin-Richard, Victor Picheny

► **To cite this version:**

Tatiana Labopin-Richard, Victor Picheny. Sequential design of experiments for estimating percentiles of black-box functions. *Statistica Sinica*, 2018, 28 (2), pp.853-877. 10.5705/ss.202016.0160 . hal-01316263v3

HAL Id: hal-01316263

<https://hal.science/hal-01316263v3>

Submitted on 19 May 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

SEQUENTIAL DESIGN OF EXPERIMENTS FOR ESTIMATING PERCENTILES OF BLACK-BOX FUNCTIONS

T. LABOPIN-RICHARD AND V. PICHENY

ABSTRACT. Estimating percentiles of black-box deterministic functions with random inputs is a challenging task when the number of function evaluations is severely restricted, which is typical for computer experiments. This article proposes two new sequential Bayesian methods for percentile estimation based on the Gaussian Process metamodel. Both rely on the Stepwise Uncertainty Reduction paradigm, hence aim at providing a sequence of function evaluations that reduces an uncertainty measure associated with the percentile estimator. The proposed strategies are tested on several numerical examples, showing that accurate estimators can be obtained using only a small number of functions evaluations.

1. INTRODUCTION

In the last decades, the question of designing experiments for the efficient exploration and analysis of numerical black-box models has received a wide interest, and metamodel-based strategies have been shown to offer efficient alternatives in many contexts, such as optimization or uncertainty quantification. We consider here the question of estimating percentiles of the output of a black-box model, with the help of Gaussian Process (GP) metamodels and sequential sampling. More precisely, let $g : \mathbb{X} \subset \mathbb{R}^d \rightarrow \mathbb{R}$ denote the output of interest of the model, the inputs of which can vary within \mathbb{X} . We assume here that the multivariate input X is modelled as a random vector; then, our objective is to estimate a percentile of $g(X)$:

$$(1) \quad q_\alpha(g(X)) = q_\alpha(Y) = F_Y^{-1}(\alpha),$$

for a fixed level $\alpha \in]0, 1[$, where $F_U^{-1} := \inf\{\mathbf{x} : F_U(\mathbf{x}) \geq u\}$ denotes the generalized inverse of the cumulative distribution function of a random variable U . We consider here only random vectors X and functions g regular enough to have $F_Y(F_Y^{-1}(\alpha)) = \alpha$ (that is, F_Y is continuous). Since the level α is fixed, we omit the index in the sequel.

A natural idea to estimate a percentile consists in using its empirical estimator: having at hand a sample $(X_i)_{i=1, \dots, n}$ of the input law X , we run it through the computer model to obtain a sample $(Y_i)_{i=1, \dots, n}$ of the output Y . Then, denoted $Y_{(k)}$ the k -th order statistic of the previous sample, the estimator

$$(2) \quad q_n := Y_{(\lfloor n\alpha \rfloor + 1)},$$

is consistent and asymptotically Gaussian under weak assumptions on the model (see [6] for more details). However, for computationally expensive models, the sample size is drastically limited, which makes the estimator (2) impractical. In that case, one may replace the sample $(X_i)_i$ by a sequence of well-chosen points that provide a useful

information for the percentile estimation. Besides, if the points X_i are not sampled with the distribution of X , the empirical percentile (2) is biased, so another estimator must be used.

In [3], the authors proposed an estimator based on the variance reduction or on the controlled stratification and give asymptotic results. Nevertheless, the most usual approach of this problem is a Bayesian method which consists in assuming that g is the realization of a well-chosen Gaussian process. In this context, [16] propose a two-step strategy: first, generate an initial set of observations to train a GP model and obtain a first estimator of the percentile, then increase the set of observations by a second set likely to improve the estimator. In [12], the authors proposed a sequential method (called GPQE and GPQE+ algorithms), based on the GP-UCB algorithm of [8], that is, making use of the confidence bounds provided by the Gaussian Process model.

In this paper we propose two new algorithms based on *Stepwise Uncertainty Reduction* (SUR), a framework that has been successfully applied to closely related problem such as optimization [17], or the dual problem of percentile estimation, the estimation of a probability of exceedance [2, 4]. A first strategy has been proposed for the percentile case in [1] and [11] that rely on expensive simulation procedures. Nevertheless, finding a statistically sound algorithm with a reasonable cost of computation, in particular when the problem dimension increases, is still an open problem.

The rest of the paper is organized as follow. In Section 2, we introduce the basics of Gaussian Process modelling, our percentile estimator and the Stepwise Uncertainty Reduction framework. Section 3 describes our two algorithms to estimate a percentile. Some numerical simulations to test the two methods are presented in Section 4, followed by concluding comments in Section 5. Most of the proofs are deferred to the Appendix.

2. GAUSSIAN PROCESS MODEL

2.1. Model definition

We consider here the classical Gaussian Process framework in computer experiments [20, 21, 18]: we suppose that g is the realization of a GP denoted by $G(\cdot)$ with known mean μ and covariance function c .

Given an observed sample $\mathcal{A}_n = \{(\mathbf{x}_1, g_1), (\mathbf{x}_2, g_2), \dots, (\mathbf{x}_n, g_n)\}$ with all $\mathbf{x}_i \in \mathbb{X}$ and $g_i = g(x_i)$, the distribution of $G|\mathcal{A}_n$ is entirely known:

$$\mathcal{L}(G|\mathcal{A}_n) = GP(m_n(\cdot), k_n(\cdot, \cdot)),$$

with, $\forall \mathbf{x} \in \mathbb{X}$,

$$\begin{aligned} m_n(\mathbf{x}) &= \mathbb{E}(G(\mathbf{x})|\mathcal{A}_n) = c_n(\mathbf{x})^T C_n^{-1} \mathbf{g}_n, \\ k_n(\mathbf{x}, \mathbf{x}') &= \text{Cov}(G(\mathbf{x}), G(\mathbf{x}')|\mathcal{A}_n) = c(\mathbf{x}, \mathbf{x}') - c_n(\mathbf{x})^T C_n^{-1} c_n(\mathbf{x}'), \end{aligned}$$

where we denote $c_n(\mathbf{x}) = [c(\mathbf{x}_1, \mathbf{x}), \dots, c(\mathbf{x}_n, \mathbf{x})]^T$, $C_n = [c(\mathbf{x}_i, \mathbf{x}_j)]_{1 \leq i, j \leq n}$ and $\mathbf{g}_n = [g_1, \dots, g_n]$. In the sequel, we also denote $s_n^2(\mathbf{x}) = k_n(\mathbf{x}, \mathbf{x})$.

We use here the standard *Universal Kriging* framework, where the covariance function depends on unknown parameters that are inferred from \mathcal{A}_n , using maximum likelihood estimates for instance. Usually, the estimates are used as face value, but updated when new observations are added to the model.

In the sequel, we will need the following property of the kriging model (see [5]).

Proposition 2.1. *Moments at step $n + 1$ are linked to the moments at step n by the one-step update formula:*

$$\left\{ \begin{array}{l} m_{n+1}(\mathbf{x}) = m_n(\mathbf{x}) + \frac{k_n(\mathbf{x}_{n+1}, \mathbf{x})}{s_n^2(\mathbf{x}_{n+1})} (g_{n+1} - m_n(\mathbf{x}_{n+1})) \\ s_{n+1}^2(\mathbf{x}) = s_n^2(\mathbf{x}) - \frac{k_n^2(\mathbf{x}_{n+1}, \mathbf{x})}{s_n^2(\mathbf{x}_{n+1})} \\ k_{n+1}(\mathbf{x}, \mathbf{x}') = k_n(\mathbf{x}, \mathbf{x}') - \frac{k_n(\mathbf{x}_{n+1}, \mathbf{x})k_n(\mathbf{x}_{n+1}, \mathbf{x}')}{s_n(\mathbf{x}_{n+1})^2} \end{array} \right. ,$$

where $(\mathbf{x}_{n+1}, g_{n+1})$ is a new observational event.

2.2 Percentile estimation

Since each call to the code g is expensive, the sequence of inputs to evaluate, $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, must be chosen carefully to make our estimator as accurate as possible. The general scheme based on GP modelling is of the following form:

- For an initial budget N_0 , we build an initialisation sample $(\mathbf{x}_0^i, g(\mathbf{x}_0^i))_{i=1 \dots N_0}$, typically using a space-filling strategy, and compute the estimator of the percentile q_{N_0} .
- At each step $n + 1 \geq N_0 + 1$ and until the budget N of evaluations is reached: knowing the current set of observations \mathcal{A}_n and estimator q_n , we choose the next point to evaluate \mathbf{x}_{n+1}^* , based on a so-called *infill criterion*. We evaluate $g(\mathbf{x}_{n+1}^*)$ and update the observations \mathcal{A}_{n+1} and the estimator q_{n+1} .
- q_N is the estimator of the percentile to return.

In the following, we describe first the estimator we choose, then the sequential strategy adapted to the estimator.

Percentile estimator. First, from a GP model we extract a percentile estimator. Considering that, conditionally on \mathcal{A}_n , the best approximation of $G(\mathbf{x})$ is $m_n(\mathbf{x})$, an intuitive estimator is simply the percentile of the GP mean:

$$(3) \quad q_n := q(m_n(X)) = q(\mathbb{E}[G(X)|\mathcal{A}_n]).$$

This is the estimator chosen for instance in [16].

Another natural idea can be to consider the estimator that minimizes the mean square error $\mathbb{E}((q - q_n)^2)$ among all \mathcal{A}_n -measurable estimator:

$$(4) \quad q_n = \mathbb{E}(q(G(X))|\mathcal{A}_n).$$

This estimator is used for instance in [12]. Despite its theoretical qualities, this estimator suffers from a major drawback, as it cannot be expressed in a computationally tractable form, and must be estimated using simulation techniques, by drawing several trajectories of $G(\cdot)$, computing the percentile of each trajectory and averaging.

Hence, in the sequel, we focus on the estimator (3), which allows us to derive closed-form expressions of its update when new observations are obtained, as we show in the next section.

Remark 2.1. *In the case of the dual problem of the probability of failure estimation $u(g) = \mathbb{P}(g(X) > u)$, this later estimator is easier to compute. Indeed, it is shown in [2]:*

$$\mathbb{E}(u(G)|\mathcal{A}_n) = \mathbb{E}\left(\int_{\mathbb{X}} \mathbf{1}_{G>u} dP_X\right) = \int_{\mathbb{X}} p_n dP_X,$$

where $p_n(\mathbf{x}) = \Phi\left(\frac{m_n(\mathbf{x})-u}{s_n(\mathbf{x})}\right)$, for Φ the cumulative distribution function of the standard Gaussian distribution. This compact form is due to the possibility to swap the integral and expectation, which is not feasible for the percentile case.

Sequential sampling and Stepwise Uncertainty Reduction. We focus here on methods based on the sequential maximization of an infill criterion, that is, of the form:

$$(5) \quad \mathbf{x}_{n+1}^* = \operatorname{argmax}_{\mathbf{x}_{n+1} \in \mathbb{X}} J_n(\mathbf{x}_{n+1}),$$

where J_n is a function that depends on \mathcal{A}_n (through the GP conditional distribution) and q_n .

Intuitively, an efficient strategy would explore \mathbb{X} enough to obtain a GP model reasonably accurate everywhere, but also exploit previous results to identify the area with response values close to the percentile and sample more densely there.

To this end, the concept of Stepwise Uncertainty Reduction (SUR) has been proposed originally in [9] as a trade-off between exploitation and exploration, and has been successfully adapted to optimization [23, 17] or probability of failure estimation frameworks [2, 4]. The general principle of SUR strategies is to define an uncertainty measure related to the objective pursued, and add sequentially the observation that will reduce the most this uncertainty. The main difficulty of such an approach is to evaluate the potential impact of a candidate point \mathbf{x}_{n+1} without having access to $g(\mathbf{x}_{n+1}) = g_{n+1}$ (that would require running the computer code).

In the percentile estimation context, [11] and [1] proposed to choose the next point to evaluate as the minimizer of the conditional variance of the percentile estimator (4). This strategy showed promising results, as it substantially outperforms pure exploration, and, in small dimension, manages to identify the percentile area (that is, where f is close to its percentile) and choose the majority of the points in it. Nevertheless, computing $V_n(\mathbf{x})$ in [11] or [1] is very costly, as it requires drawing many GP realizations, which hinders its use in practice for dimensions larger than two.

In next section, we propose other functions J that also quantify the uncertainty associated with our estimator, but which have closed forms and then are less expensive to compute. To do so, we first exhibit the formula to update the current estimator q_n (build on \mathcal{A}_n) to $q_{n+1}(\mathbf{x}_{n+1})$ the estimator at step $n+1$ if we had chosen $\mathbf{x}_{n+1}^* = \mathbf{x}_{n+1}$.

3. MAIN RESULTS

3.1. Update formula for the percentile estimator

In this section, we express the estimator $q_{n+1}(\mathbf{x}_{n+1})$ as a function of the past observations \mathcal{A}_n , the past percentile estimator q_n , a candidate point \mathbf{x}_{n+1} and its evaluation g_{n+1} .

We focus on the estimator (3), which is at step n the percentile of the random vector $m_n(X)$. Since no closed-form expression is available, we approach it by using the

empirical percentile. Let $\mathbf{X}_{\text{MC}} = (\mathbf{x}_{\text{MC}}^1, \dots, \mathbf{x}_{\text{MC}}^l)$ be an independent sample of size l , distributed as X . We compute $m_n(\mathbf{X}_{\text{MC}})$ and order this vector by denoting $m_n(\mathbf{X}_{\text{MC}})_{(i)}$ the i -th coordinate. Then we choose

$$(6) \quad q_n = m_n(\mathbf{X}_{\text{MC}})_{([\alpha]+1)}.$$

Remark 3.1. *Since the observation points $(\mathbf{x}_1, \dots, \mathbf{x}_n)$ do not follow the distribution of X , they cannot be used to estimate the percentile. Hence, a different set (\mathbf{X}_{MC}) must be used.*

In the sequel, we denote by \mathbf{x}_n^q the point of \mathbf{X}_{MC} such that

$$q_n = m_n(\mathbf{x}_n^q).$$

This point is referred to as *percentile point*.

Now, let us consider that a new observation $g_{n+1} = g(\mathbf{x}_{n+1})$ is added to \mathcal{A}_n . The key of SUR strategies is to measure the impact of this observation on our estimator q_n , that is, express $q_{n+1} = m_{n+1}(\mathbf{x}_{n+1}^q)$ as a function of g_{n+1} and \mathbf{x}_{n+1} .

First, by Proposition 2.1, we have:

$$(7) \quad m_{n+1}(\mathbf{X}_{\text{MC}}) = m_n(\mathbf{X}_{\text{MC}}) + \frac{k_n(\mathbf{X}_{\text{MC}}, \mathbf{x}_{n+1})}{s_n(\mathbf{x}_{n+1})^2} (g_{n+1} - m_n(\mathbf{x}_{n+1})).$$

We see directly that once \mathbf{x}_{n+1} is fixed, all the vector $m_{n+1}(\mathbf{X}_{\text{MC}})$ is determined by the value of g_{n+1} . Our problem is then to find, for any g_{n+1} in \mathbb{R} , which point of \mathbf{X}_{MC} is the percentile point, that is, which point satisfies

$$(8) \quad m_{n+1}(\mathbf{X}_{\text{MC}})_{([\alpha]+1)} = m_{n+1}(\mathbf{x}_{n+1}^q).$$

Let us denote $\mathbf{b} = m_n(\mathbf{X}_{\text{MC}})$ and $\mathbf{a} = k_n(\mathbf{X}_{\text{MC}}, \mathbf{x}_{n+1})$, which are vectors of \mathbb{R}^l , and $z = \frac{g_{n+1} - m_n(\mathbf{x}_{n+1})}{s_n^2(\mathbf{x}_{n+1})}$, so that the updated mean simply writes as a linear function of z , $\mathbf{b} + \mathbf{a}z$. Our problem can then be interpreted graphically: each coordinate of $m_{n+1}(\mathbf{X}_{\text{MC}})$ is represented by a straight line of equation:

$$(9) \quad b_i + a_i z, \quad i \in \{1, \dots, l\},$$

and the task of finding \mathbf{x}_{n+1}^q for any value of g_{n+1} amounts to finding the $[\alpha] + 1$ lowest line for any value of z .

We can first notice that the lines order changes only when two lines intersect each other. There are at most $L = \frac{l(l-1)}{2}$ intersection points, which we denote I_1, \dots, I_L , in increasing order. We set $I_0 = -\infty$ and $I_{L+1} = +\infty$, and introduce $(B_i)_{0 \leq i \leq L}$, the sequence of intervals between intersection points:

$$(10) \quad B_i = [I_i, I_{i+1}] \text{ for } i \in [0, L]$$

For any $z \in B_i$, the order of $(b_i + a_i z)$ is fixed.

Denoting j_i the index of the $[\alpha] + 1$ lowest line, we have:

$$(11) \quad \mathbf{x}_{n+1}^q = \mathbf{x}_{\text{MC}}^{j_i}, \quad z \in B_i,$$

the percentile point when $z \in B_i$, which we henceforth denote $\mathbf{x}_{n+1}^q(B_i)$.

Finally, we have shown that

Proposition 3.1. *Under previous notations, at step n (when we know \mathcal{A}_n, q_n), for the candidate point \mathbf{x}_{n+1} we get*

$$q_{n+1}(\mathbf{x}_{n+1}, g_{n+1}) = \sum_{i=0}^L m_{n+1}(\mathbf{x}_{n+1}^q(B_i)) \mathbf{1}_{z \in B_i}.$$

Intuitively, the updated percentile is equal to the updated GP mean at one of the \mathbf{X}_{MC} points, that depends on which interval g_{n+1} (or equivalently, z) falls.

Figure 1 provides an illustrative example of this proposition for $l = 5$, and $\alpha = 40\%$. The values of a and b are given by a GP model, which allows us to draw the straight lines (black) as a function of z . Each line corresponds to a point \mathbf{x}_{MC}^i . The intersections for which the percentile point changes are shown by the vertical lines. For each interval, the segment corresponding to the percentile point (second lowest line) is shown in bold. We see that depending on the value of z (that is, the value of g_{n+1}), the percentile point changes. On the example, j_i takes successively as values 2, 3, 1, 4, 3 and 5.

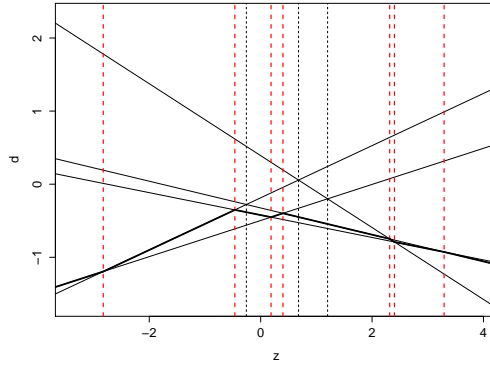


FIGURE 1. Evolution of the percentile point as a function of the value of z . Each plain line represents a point of \mathbf{X}_{MC} , and the vertical lines the relevant intersections I_i . The second lowest line is shown in bold.

Remark 3.2. *Although the number of intersections grows quadratically with the MC sample size, finding the set of percentile points can be done very efficiently, based on two important elements: first, the number of distinct percentile points is much smaller than the number of intersections (five and ten, respectively, on Figure 1, but this difference increases rapidly with l); second, the order of the straight lines remains the same except for two elements for two adjacent intervals. This latter feature allows us to avoid numerous calls to sorting functions.*

In the following, the notations B_i and L denote the effective intervals and number of intervals, respectively, that is, the intervals for which the percentile points are different.

3.2. Infill criterion based on probability of exceedance

The proposition 3.1 allows us to express the percentile estimator at step $n + 1$ as a function of the candidate point \mathbf{x}_{n+1} and corresponding value g_{n+1} . In this section, we

use this formulation to define a SUR criterion, that is, an uncertainty measure related to our estimator that can be minimized by a proper choice of \mathbf{x}_{n+1} .

This criterion is inspired from related work in probability of failure estimation [2] and multi-objective optimization [17], that take advantage of the closed-form expressions of probabilities of exceeding thresholds in the GP framework.

By definition, the percentile is related to the probability of exceedance by

$$(12) \quad \mathbb{P}(G(X) \geq q(G(X))) = 1 - \alpha.$$

Our idea is the following. The probability $\mathbb{P}(G(\mathbf{x}) \geq q_n | \mathcal{A}_n)$, available for any $\mathbf{x} \in \mathbb{X}$, is in the ideal case (G is exactly known) either zero or one, and, if $q_n = q(G(X))$, the proportion of ones is exactly equal to $1 - \alpha$. At step n , a measure of error is then:

$$(13) \quad J_n^{\text{prob}} = \left| \int_{\mathbb{X}} \mathbb{P}(G(\mathbf{x}) \geq q_n | \mathcal{A}_n) d\mathbf{x} - (1 - \alpha) \right| = |\Gamma_n - (1 - \alpha)|,$$

with $\Gamma_n = \int_{\mathbb{X}} \mathbb{P}(G(\mathbf{x}) \geq q_n | \mathcal{A}_n) d\mathbf{x}$.

Following the SUR paradigm, the point we would want to add at step $n + 1$ is the point \mathbf{x}_{n+1}^* satisfying

$$(14) \quad \mathbf{x}_{n+1}^* = \underset{\mathbf{x}_{n+1} \in \mathbb{X}}{\text{argmin}} J_{n+1}^{\text{prob}}(\mathbf{x}_{n+1}).$$

As seen in proposition 3.1, q_{n+1} , and consequently $J_{n+1}^{\text{prob}}(\mathbf{x}_{n+1})$, depend on the candidate evaluation g_{n+1} , which makes it computable only by evaluating g . To circumvent this problem, we replace g_{n+1} by its distribution conditional on \mathcal{A}_n . We can then choose the following criterion to minimize (indexed by \mathbf{x}_{n+1} to make the dependency explicit):

$$(15) \quad J_n^{\text{prob}}(\mathbf{x}_{n+1}) = |\mathbb{E}_{G_{n+1}}(\Gamma_{n+1}(x+1)) - (1 - \alpha)|$$

where now,

$$(16) \quad \Gamma_{n+1}(x_{n+1}) = \int_{\mathbb{X}} \mathbb{P}(G(\mathbf{x}) \geq q_{n+1} | A_{n+1}) d\mathbf{x},$$

with $A_{n+1} = \mathcal{A}_{n+1} \cup (\mathbf{x}_{n+1}, G_{n+1})$ and G_{n+1} is still in its random form.

We show that

Proposition 3.2. *Using previous notations and under our first strategy,*

$$J_n^{\text{prob}}(\mathbf{x}_{n+1}) = \left| \int_{\mathbb{X}} \sum_{i=1}^{L-1} \left[\Phi_{r_i^n} (e_n^i(\mathbf{x}_{n+1}; \mathbf{x}), f_n^i(\mathbf{x}_{n+1}, I_{i+1})) - \Phi_{r_i^n} (e_n^i(\mathbf{x}_{n+1}; \mathbf{x}), f_n(\mathbf{x}_{n+1}, I_i)) \right. \right. \\ \left. \left. + \Phi_{r_i^n} ((e_n^i(\mathbf{x}_{n+1}; \mathbf{x}), f_n^i(\mathbf{x}_{n+1}, I_1)) + \Phi_{-r_i^n} (e_n^i(\mathbf{x}_{n+1}; \mathbf{x}), -f_n^i((\mathbf{x}_{n+1}, I_L))) \right] d\mathbf{x} - (1 - \alpha) \right|$$

where

$$e_n^i(\mathbf{x}_{n+1}; \mathbf{x}; \mathbf{x}_{n+1}^q(B_i)) = \frac{m_n(\mathbf{x}) - m_n(\mathbf{x}_{n+1}^q(B_i))}{\sigma_W}, \quad f_n^i(\mathbf{x}_{n+1}; I_i) = I_i s_n(\mathbf{x}_{n+1}),$$

$$\sigma_W = s_n(\mathbf{x})^2 + \frac{k_n(\mathbf{x}_{n+1}^q(B_i), \mathbf{x}_{n+1})^2}{s_n(\mathbf{x}_{n+1})^2} - 2 \frac{k_n(\mathbf{x}_{n+1}^q(B_i), \mathbf{x}_{n+1})k_n(\mathbf{x}, \mathbf{x}_{n+1})}{s_n(\mathbf{x}_{n+1})^2}$$

and $\Phi_{r_n^i}$ is the cumulative distribution function (CDF) of the centered Gaussian law of covariance matrix

$$K = \begin{pmatrix} 1 & r_n^i \\ r_n^i & 1 \end{pmatrix}$$

with

$$r_n^i = \frac{k_n(\mathbf{x}_{n+1}^q(B_i), \mathbf{x}_{n+1}) - k_n(\mathbf{x}, \mathbf{x}_{n+1})}{\sqrt{s_n(\mathbf{x})^2 + \frac{k_n(\mathbf{x}_{n+1}^q(B_i), \mathbf{x}_{n+1})^2}{s_n(\mathbf{x}_{n+1})^2} - 2 \frac{k_n(\mathbf{x}_{n+1}^q(B_i), \mathbf{x}_{n+1})k_n(\mathbf{x}, \mathbf{x}_{n+1})}{s_n(\mathbf{x}_{n+1})^2} s_n(\mathbf{x}_{n+1})}}.$$

The proof is deferred to the Appendix.

Despite its apparent complexity, this criterion takes a favourable form, since it writes as a function of GP quantities at step n (m_n , s_n and k_n), which can be computed very quickly once the model is established. Besides, it does not require conditional simulations (as the criterion in [12]), which is a decisive advantage both in terms of computational cost and evaluation precision.

Let us stress here, however, that evaluating this criterion requires a substantial computational effort, as it takes the form of an integral over \mathbb{X} , which must be done numerically. An obvious choice here is to use the set \mathbf{X}_{MC} as integration points. Also, it relies on the bivariate Gaussian CDF, which also must be computed numerically. Very efficient programs can be found, such as the R package `pbivnorm` [13], which makes this task relatively inexpensive.

3.3. Infill criterion based on the percentile variance

Accounting for the fact that, although not relying on conditional simulations, J^{prob} is still expensive to compute, we propose here an alternative, that does not require numerical integration over \mathbb{X} .

Since we want q_n to converge to the quantile, it is important that this estimator becomes increasingly *stable*. The variance of $q_{n+1}|A_{n+1}$ is a good indicator of this stability, as it gives the fluctuation range of q_{n+1} as a function of the different possible values of g_{n+1} . However, choosing the point that minimizes at each step n this variance has no sense here, as choosing $\mathbf{x}_{n+1} \in \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ (that is, duplicating an existing observation) would result in $\text{Var}(q_{n+1}|A_{n+1}) = 0$.

Inversing the SUR paradigm, we propose to choose the point that *maximizes* this variance. By doing so, we will obtain the sequence of points which evaluations have a large impact on the estimator value, hence reducing sequentially the instability of our estimator:

$$(17) \quad J_n^{\text{Var}}(\mathbf{x}_{n+1}) = \text{Var}_{G_{n+1}}(q_{n+1}|A_{n+1})$$

where once again A_{n+1} denotes the conditioning on $\mathcal{A}_n \cup (\mathbf{x}_{n+1}, G_{n+1})$, with G_{n+1} random. We can show that:

Proposition 3.3. *Using the previous notations, conditionally on \mathcal{A}_n and on the choice of \mathbf{x}_{n+1} :*

$$\begin{aligned} J_n^{Var}(\mathbf{x}_{n+1}) &= \sum_{i=1}^L [k_n(\mathbf{x}_{n+1}^q(B_i), \mathbf{x}_{n+1})]^2 V(s_n(\mathbf{x}_{n+1}), I_{i+1}, I_i) P_i \\ &+ \sum_{i=1}^L [m_n(\mathbf{x}_{n+1}^q(B_i) - k_n(\mathbf{x}_{n+1}^q(B_i), \mathbf{x}_{n+1}) E(s_n(\mathbf{x}_{n+1}), I_{i+1}, I_i)]^2 (1 - P_i) P_i \\ &- 2 \sum_{i=2}^L \sum_{j=1}^{i-1} [m_n(\mathbf{x}_{n+1}^q(B_i) - k_n(\mathbf{x}_{n+1}^q(B_i), \mathbf{x}_{n+1}) E(s_n(\mathbf{x}_{n+1}), I_{i+1}, I_i)] P_i \\ &\quad [m_n(\mathbf{x}_{n+1}^q(B_j) - k_n(\mathbf{x}_{n+1}^q(B_j), \mathbf{x}_{n+1}) E(s_n(\mathbf{x}_{n+1}), I_{j+1}, I_j)] P_j, \end{aligned}$$

where:

$$\begin{aligned} P_i &= \Phi(s_n(\mathbf{x}_{n+1}) I_{i+1}) - \Phi(s_n(\mathbf{x}_{n+1}) I_i), \\ E(s_n(\mathbf{x}_{n+1}), I_{i+1}, I_i) &= \frac{1}{s_n(\mathbf{x}_{n+1})} \left(\frac{\phi(s_n(\mathbf{x}_{n+1}) I_{i+1}) - \phi(s_n(\mathbf{x}_{n+1}) I_i)}{\Phi(s_n(\mathbf{x}_{n+1}) I_{i+1}) - \Phi(s_n(\mathbf{x}_{n+1}) I_i)} \right), \end{aligned}$$

and

$$\begin{aligned} V(s_n(\mathbf{x}_{n+1}), I_{i+1}, I_i) &= \\ &= \frac{1}{s_n(\mathbf{x}_{n+1})^2} \left[1 + \frac{s_n(\mathbf{x}_{n+1}) \phi(I_{i+1}) - s_n(\mathbf{x}_{n+1}) \phi(I_i)}{\Phi(I_{i+1}) - \Phi(I_i)} - \left(\frac{\phi(I_{i+1}) - \phi(I_i)}{\Phi(I_{i+1}) - \Phi(I_i)} \right)^2 \right], \end{aligned}$$

for Φ and ϕ respectively the cumulative distribution function and density function of the standard Gaussian law.

The proof is deferred to Appendix.

Again, this criterion writes only as a function of GP quantities at step n (m_n , s_n and k_n). As it does not require numerical integration nor the bivariate CDF, it is considerably cheaper to compute than the previous one.

Figure 2 provides an illustration of the concepts behind this criterion, by showing how different values of g_{n+1} affect the estimator. Here, one updated mean is drawn for each interval B_i (that is, with z taking its value in the middle of the interval). The corresponding 90% percentiles, as well as the percentile points \mathbf{x}_{n+1}^q vary substantially, depending on g_{n+1} , which results in a large variance $\text{Var}(g_{n+1} | \mathcal{A}_{n+1})$. Hence, the point $\mathbf{x}_{n+1} = 0.9$ can be considered as highly informative for our estimator.

4. NUMERICAL SIMULATIONS

4.1. Two-dimensional example

As an illustrating example, we use here the classical Branin test function (see [7] Equation 19 in Appendix). On $[0, 1]^2$, the range of this function is approximately $[0, 305]$.

We take: $X_1, X_2 \sim \mathcal{U}[0, 1]$, and search for the 85% percentile. The initial set of experiments consists of seven observations generated using Latin Hypercube Sampling (LHS), and 11 observations are added sequentially using both SUR strategies. The GP models learning, prediction and update is performed using the R package `DiceKriging` [19]. The covariance is chosen as Matérn 3/2 and the mean as a linear trend.

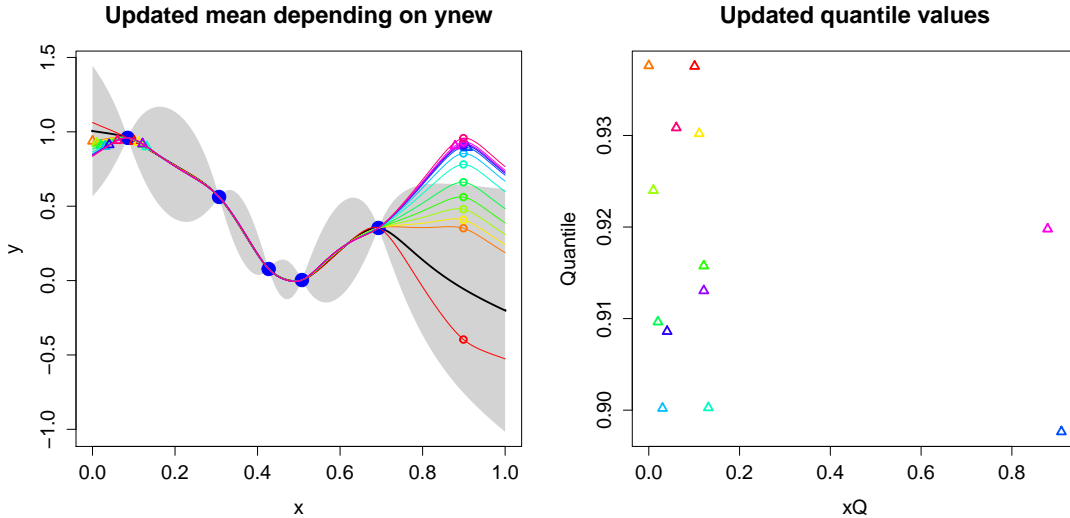


FIGURE 2. Illustration of the J^{Var} criterion. Left: GP model (black bold line and grey area) and updated GP mean (other lines) depending on the value of g_{new} (circles) for $\mathbf{x}_{n+1} = 0.9$. The corresponding 90% percentiles q_{n+1} are shown with the triangles. Right: percentile values only, indexed by the corresponding percentile points.

For \mathbf{X}_{MC} , we used a 1000-point uniform sample on $[0, 1]^2$. For simplicity purpose, the search of \mathbf{x}_{n+1} is performed on \mathbf{X}_{MC} , although a continuous optimizer algorithm could have been used here. The actual percentile is computed using a 10^5 -point sample.

Figure 4 reports the final set of experiments, along with contour lines of the GP model mean, and Figure 5 the evolution of the estimators. In addition, Figure 3 shows three intermediate stages of the J_n^{Var} run.

Figure 3 reveals the dynamic of our strategy: from the initial design of experiments, the top right corner of the domain is identified as the region containing the highest 15% values. Several observations are added in that region until the kriging approximation becomes accurate, then a new region (bottom left corner) is explored (square point, Figure 3 right).

The two strategies lead to relatively similar observation sets (Figure 4), that mostly consist of values close to the contour line corresponding to the 85th percentile (exploitation points), and a couple of space-filling points (exploration points). With 18 observations, both estimators are close to the actual value (in particular with respect to the range of the function), yet additional observations may be required to achieve convergence (Figure 5).

4.2. Four and six dimensional examples

We consider now two more difficult test functions, with four and six dimensions, respectively (*hartman* and *ackley* functions, see Equations 20 and 21 in Appendix). Both are widely used to test optimization strategies [7], and are bowl-shaped, multi-modal functions.

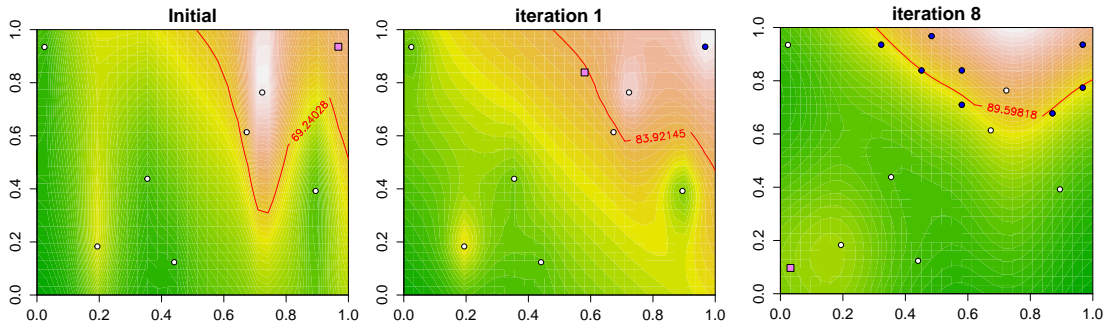


FIGURE 3. Contour lines of the GP mean and experimental set at three different n values (7, 8, and 15) with the J_n^{Var} criterion. The initial observations are shown with white circles, the observations added by the sequential strategy with blue circles, and the next point to evaluate with violet squares. The line shows the contour corresponding to the percentile estimate.

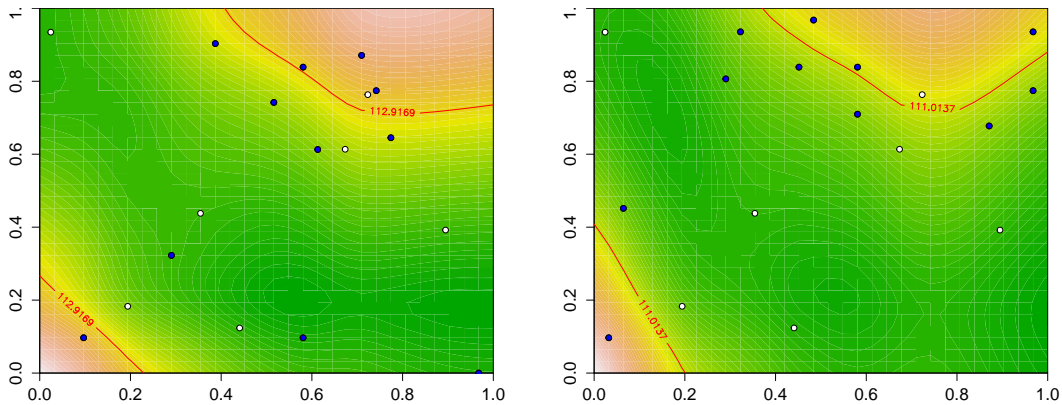


FIGURE 4. Comparison of observation sets obtained using J_n^{prob} (left) and J_n^{Var} (right).

We take on both cases: $\mathbf{X} \sim \mathcal{N}(\frac{1}{2}, \Sigma)$, with Σ a symmetric matrix with diagonal elements equal to 0.1 and other elements equal to 0.05. The initial set of observations is taken as a 30-point LHS, and 60 observations are added sequentially. A 3000-point sample from the distribution of \mathbf{X} is used for \mathbf{X}_{MC} (renewed at each iteration), and the actual percentile is computed using a 10^5 -point sample. Again, the GP covariance is chosen as Matérn 3/2 and the mean as a linear trend.

The criteria are optimized as follow: a (large) set of 10^5 candidates is generated from the distribution of \mathbf{X} , out of which a shorter set of 300 “promising” points is extracted. Those points are drawn randomly from the large set with weights equal to $\phi\left(\frac{q_n - m_n(\mathbf{x})}{s_n(\mathbf{x})}\right)$. Hence, higher weights are given to points either close to the current estimate and/or with high uncertainty. The criterion is evaluated on this subset of points and the best is

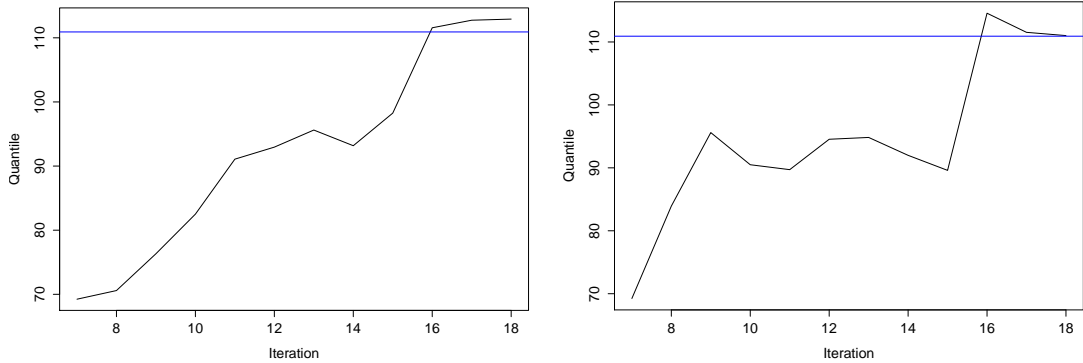


FIGURE 5. Evolution of the percentile estimates using J_n^{prob} (left) and J_n^{Var} (right) for the 2D problem. The horizontal line shows the actual 85th percentile.

chosen as the next infill point. In addition, for J_n^{Var} a local optimization is performed, starting from the best point of the subset (using the BFGS algorithm, see [15]). Due to computational constraints, this step is not applied to J_n^{prob} , which is more costly. However, preliminary experiments have shown that only a limited gain is achieved by this step.

As an baseline strategy for comparison purpose, we also include a “random search”, that is, the \mathbf{x}_{n+1} are sampled randomly from the distribution of X .

Several percentile levels are considered in order to cover a variety of situations: 5% and 97% for the 4D problem and 15% and 97% for the 6D problem. Due to the bowl-shape of the functions, low levels are defined by small regions close to the center of the support of X , while high levels correspond to the edges of the support of X . Besides, it is reasonable to assume that levels farther away from 50% are more difficult to estimate.

As an error metric ε , we consider the absolute difference between the percentile estimator and its actual value. We show this error as a percentage of the variation range of the test function. Since X is not bounded, the range is defined as the difference between the 0.5 and 99.5 percentiles of $g(X)$.

To assess the robustness of our approach, the experiments are run ten times for each case, starting with a different initial set of observations. The evolution of the estimators (average, lowest and highest error metric values over the ten runs) is given in Figures 6.

First of all, we see that except on one case (4D, $\alpha = 0.97$ and J_n^{prob}), on average both strategies provide estimates with less than 2% error after approximately 30 iterations (for a total of 60 function evaluations), which plainly justifies the use of GP models and sequential strategies in a constrained budget context.

For $d = 4$, $\alpha = 0.05$, both methods seem to converge to the actual percentile, J_n^{prob} being slightly better, in particular in terms of consistency and for the latest steps.

For $d = 4$, $\alpha = 0.97$, J_n^{Var} reaches very quickly for all runs a good estimate (less than 1% error), yet seems to converge then slowly to the exact solution. This might be explained by the relative mismatch between the GP model and the test function. J_n^{prob}

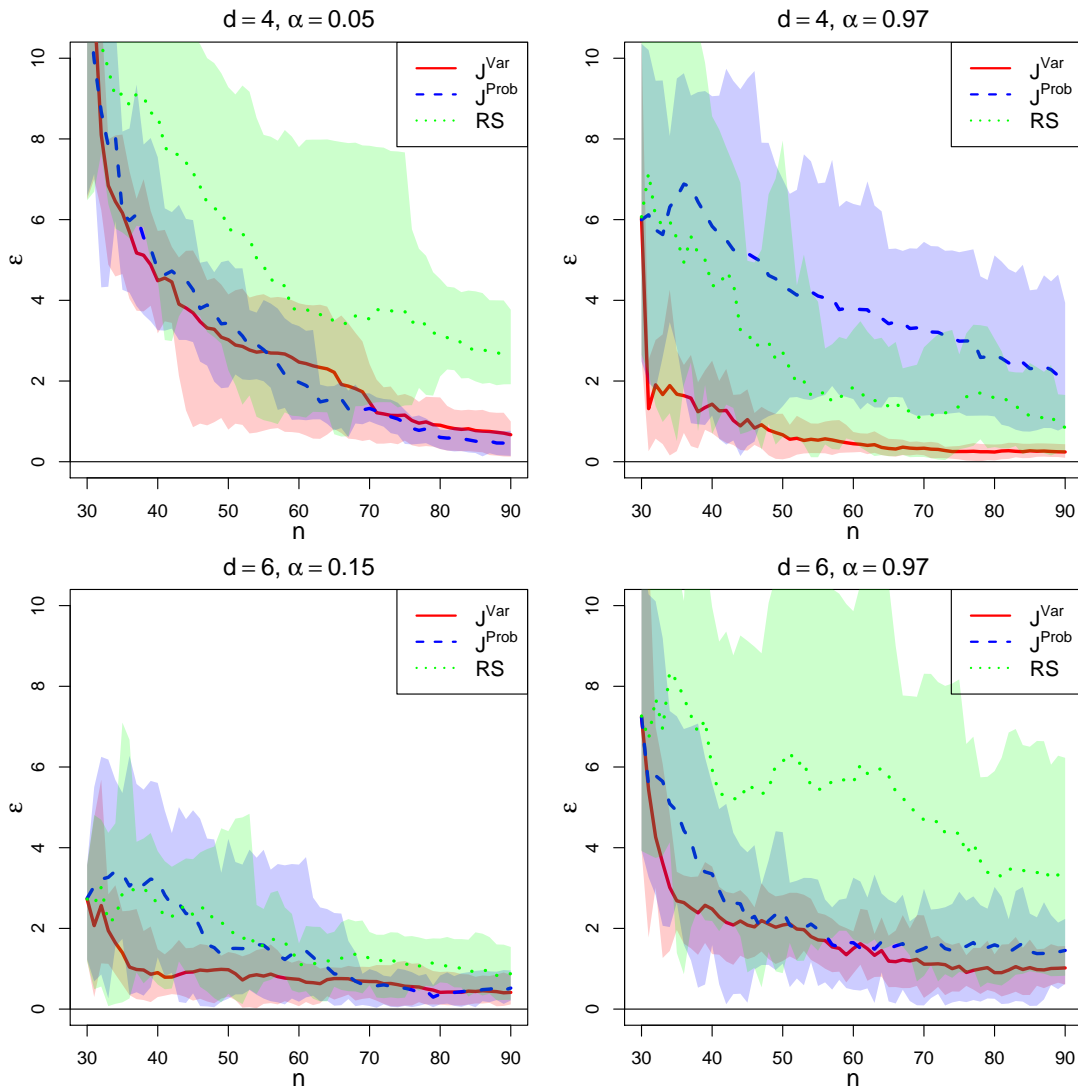


FIGURE 6. Evolution of the percentile estimates using J_n^{prob} (dashed line), J_n^{Var} (plain line) or random search (RS, dotted line) for the 4D and 6D problems and several percentile levels. The lines show the average error and the shaded areas the 10% and 90% quantile errors over the runs.

performs surprisingly poorly; we conjecture that a more exploratory behavior compared to J_n^{Var} hinders its performance here.

For $d = 6$, $\alpha = 0.15$, both approaches reach consistently less than 1% error. However, they outperform only moderately the random search strategy here. This might indicate that for central percentile values, less gain can be achieved by sequential strategies, as a large region of the design space needs to be learned to characterize the percentile, making space-filling strategies for instance competitive.

Finally, for $d = 6$, $\alpha = 0.97$, both approaches largely outperform random search, yet after a first few very efficient steps seem to converge only slowly to the actual percentile.

In general, those experiments show the ability of our approach to handle multi-modal black-box functions, with input space dimensions typical of GP-based approaches. Our results seem to indicate a better efficiency of the J_n^{Var} criterion, yet the better convergence with J_n^{prob} for $d = 4$, $\alpha = 0.05$ might call for hybrid strategies, with early steps performed with J_n^{Var} and late steps with J_n^{prob} .

5. CONCLUDING COMMENTS

We have proposed two efficient sequential Bayesian strategies for percentile estimation. Both approaches rely on the analytical update formula for the GP-based estimator, which has been obtained thanks to the particular form of the GP equations and the introduction of the *quantile point* concept. Two criteria have then been proposed based either on probability of exceedance or on variance, for which closed-form expression have been derived, hence avoiding the use of computationally intensive conditional simulations. Numerical experiments in dimensions two to six have demonstrated the potential of both approaches.

There are of course some limitations of the proposed method, that call for future improvements. Both strategies rely on the set \mathbf{X}_{MC} , which size is in practice limited by the computational resources to a couple of thousands at most. This may hinder the use of our method for extreme percentile estimation, or for highly multi-modal functions. Combining adaptive sampling strategies or subset selection methods with our approaches may prove useful in this context.

Accounting for the GP model error (due to an inaccurate estimation of its hyperparameters or a poor choice of kernel) is also an important task, that may improve greatly the efficiency and robustness of the approach. Embracing a fully Bayesian approach (as for instance in [14, 10]) may help address this issue, yet at the price of additional computational expense.

ACKNOWLEDGEMENTS

The authors would like to thank Damien Leroux for his help on the efficient implementation of the algorithms on R.

6.1. Proof of Proposition 3.2

In the following, we denote \mathbb{E}_n and \mathbb{P}_n the expectation and the probability conditionally on the event \mathcal{A}_n . Starting from Equation 16, we have:

$$\begin{aligned} \mathbb{E}_{G_{n+1}}(\Gamma_{n+1}(\mathbf{x}_{n+1})) &= \mathbb{E}_{G_{n+1}} \left[\int_{\mathbb{X}} \mathbb{P}(G(x) \geq q_{n+1}) | \mathcal{A}_{n+1} dx \right] \\ &= \int_{\mathbb{X}} \mathbb{E}_{G_{n+1}} \left[\mathbb{E}_n [\mathbf{1}_{G(x) \geq q_{n+1}(x_{n+1})} | G_{n+1}] \right] dx \\ &= \int_{\mathbb{X}} \mathbb{E}_n [\mathbf{1}_{G(x) \geq q_{n+1}(\mathbf{x}_{n+1})}] dx \\ &= \int_{\mathbb{X}} \mathbb{P}_n(G(x) \geq q_{n+1}(\mathbf{x}_{n+1})) dx . \end{aligned}$$

We get then:

$$J_{n+1}^{\text{prob}}(\mathbf{x}_{n+1}) = \left| \int_{\mathbb{X}} \mathbb{P}_n(G(x) \geq q_{n+1}(\mathbf{x}_{n+1})) dx - (1 - \alpha) \right|.$$

Now, to get an closed form of our criterion, we have to develop $\mathbb{P}_n(G(x) \geq q_{n+1}(x_{n+1}))$. To do so, we use Proposition 3.1. Denoting $Z = \frac{G_{n+1} - m_n(\mathbf{x}_{n+1})}{s_n(\mathbf{x}_{n+1})^2}$, we have:

$$\begin{aligned} \mathbb{E}_n(\mathbf{1}_{G(x) \geq q_{n+1}(\mathbf{x}_{n+1})}) &= \sum_{i=0}^L \mathbb{E}_n \left[\mathbf{1}_{G(x) \geq m_{n+1}(\mathbf{x}_{n+1}^q(B_i))} \mathbf{1}_{Z \in B_i} \right] \\ &= \sum_{i=1}^{L-1} \left(\mathbb{P}_n \left[G(x) \geq m_{n+1}(\mathbf{x}_{n+1}^q(B_i)) \cap Z \leq I_{i+1} \right] \right. \\ &\quad \left. - \mathbb{P}_n \left[G(x) \geq m_{n+1}(\mathbf{x}_{n+1}^q(B_i)) \cap Z \leq I_i \right] \right) \\ &\quad + \mathbb{P}_n(G(x) \geq m_{n+1}(\mathbf{x}_{n+1}^q(B_1)) \cap Z \leq I_1) \\ &\quad + \mathbb{P}_n(G(x) \geq m_{n+1}(\mathbf{x}_{n+1}^q(B_L)) \cap Z \geq I_L). \end{aligned}$$

Now,

$$\begin{aligned} T_n &:= \mathbb{P}_n(G(x) \geq m_{n+1}(\mathbf{x}_{n+1}^q(B_i)) \cap Z \leq I_i) \\ &= \mathbb{P}_n(m_{n+1}(\mathbf{x}_{n+1}^q(B_i)) - G(x) \leq 0 \cap Z \leq I_i), \end{aligned}$$

is the cumulative distribution function of the couple $(m_{n+1}(\mathbf{x}_{n+1}^q(B_i)) - G(x), Z) := (W, Z)$, at point $(0, I_i)$. This random vector, conditionally on \mathcal{A}_n is Gaussian. We denote by M and R its mean vector and covariance matrix, respectively.

Thanks to Proposition 2.1, we have:

$$(18) \quad m_{n+1}(\mathbf{x}_{n+1}^q(B_i)) = m_n(\mathbf{x}_{n+1}^q(B_i)) - k_n(\mathbf{x}_{n+1}^q(B_i), \mathbf{x}_{n+1})Z,$$

which gives

$$M = \begin{pmatrix} m_n(\mathbf{x}_{n+1}^q(B_i)) - m_n(x) \\ 0 \end{pmatrix}, \quad R = \begin{pmatrix} \text{Var}(W) & \text{Cov}(W, Z) \\ \text{Cov}(W, Z) & \text{Var}(Z) \end{pmatrix},$$

with

$$\text{Var}(W) := \sigma_W = s_n(x)^2 + \frac{k_n(\mathbf{x}_{n+1}^q(B_i), \mathbf{x}_{n+1})^2}{s_n(\mathbf{x}_{n+1})^2} - 2 \frac{k_n(\mathbf{x}_{n+1}^q(B_i), \mathbf{x}_{n+1})k_n(x, \mathbf{x}_{n+1})}{s_n(\mathbf{x}_{n+1})^2},$$

$$\text{Cov}(W, Z) = \frac{k_n(\mathbf{x}_{n+1}^q(B_i), \mathbf{x}_{n+1}) - k_n(x, \mathbf{x}_{n+1})}{s_n(\mathbf{x}_{n+1})^2} \quad \text{and} \quad \text{Var}(Z) = \frac{1}{s_n(\mathbf{x}_{n+1})^2}.$$

We can conclude by centering and normalizing:

$$\begin{aligned} T_n &= \mathbb{P}_n(W \leq 0 \cap Z \leq I_i) \\ &= \mathbb{P} \left(\frac{W - (m_n(\mathbf{x}_{n+1}^q(B_i)) - m_n(x))}{\sqrt{\text{Var}(W)}} \leq \frac{m_n(x) - m_n(\mathbf{x}_{n+1}^q(B_i))}{\sqrt{\text{Var}(W)}} \cap s_n(\mathbf{x}_{n+1})Z \leq I_i s_n(\mathbf{x}_{n+1}) \right) \\ &:= \mathbb{P}(S \leq e_n^i(\mathbf{x}_{n+1}; x; \mathbf{x}_{n+1}^q(B_i)) \cap T \leq f_n^i(\mathbf{x}_{n+1}; I_i)), \end{aligned}$$

where (S, T) is a Gaussian random vector of law $\mathcal{N}\left(0, \begin{pmatrix} 1 & r_n^i \\ r_n^i & 1 \end{pmatrix}\right)$ with

$$r_n^i := r_n(\mathbf{x}_{n+1}; x; \mathbf{x}_{n+1}^q(B_i)) = \frac{k_n(\mathbf{x}_{n+1}^q(B_i), \mathbf{x}_{n+1}) - k_n(x, \mathbf{x}_{n+1})}{\sqrt{\text{Var}(W)}s_n(\mathbf{x}_{n+1})} .$$

$$e_n^i(\mathbf{x}_{n+1}; x; \mathbf{x}_{n+1}^q(B_i)) = \frac{m_n(x) - m_n(\mathbf{x}_{n+1}^q(B_i))}{\sqrt{\text{Var}(W)}} ,$$

and

$$f_n^i(\mathbf{x}_{n+1}; I_i) = I_i s_n(\mathbf{x}_{n+1}) .$$

Finally, we get for $1 \leq i \leq L$,

$$\mathbb{P}_n(G(x) \geq m_{n+1}(\mathbf{x}_{n+1}^q(B_i)) \cap Z \leq I_i) = \Phi_{r_n^i}(e_n^i(\mathbf{x}_{n+1}; x; \mathbf{x}_{n+1}^q(B_i)), f_n^i(\mathbf{x}_{n+1}; I_i)) ,$$

where we denote Φ_r the cumulative distribution function of the centered Gaussian random vector of covariance matrix

$$\Sigma = \begin{pmatrix} 1 & r \\ r & 1 \end{pmatrix} .$$

Similarly, for $0 \leq i \leq L$:

$$\mathbb{P}_n(G(x) \geq m_{n+1}(\mathbf{x}_{n+1}^q(B_i)) \cap Z \leq I_{i+1}) = \Phi_{r_n^i}(e_n^i(\mathbf{x}_{n+1}; x; \mathbf{x}_{n+1}^q(B_i)), f_n^i(\mathbf{x}_{n+1}; I_{i+1})) ,$$

and

$$\mathbb{P}_n(G(x) \geq m_{n+1}(\mathbf{x}_{n+1}^q(B_i)) \cap Z \geq I_L) = \Phi_{-r_n^i}(e_n^i(\mathbf{x}_{n+1}; x; \mathbf{x}_{n+1}^q(B_i)), -f_n^i(\mathbf{x}_{n+1}; I_L)) .$$

6.2. Proof of Proposition 3.3

We first recall the following total variance law formula:

Lemma 5.1. *Let E_1, \dots, E_n be mutually exclusive and exhaustive events. Then, for a random variable U , the following equality holds:*

$$\begin{aligned} \text{Var}(U) &= \sum_{i=1}^n \text{Var}(U | E_i) \mathbb{P}(E_i) + \sum_{i=1}^n \mathbb{E}(U | E_i)^2 (1 - \mathbb{P}(E_i)) \mathbb{P}(E_i) \\ &\quad - 2 \sum_{i=2}^n \sum_{j=1}^{i-1} \mathbb{E}(U | E_i) \mathbb{P}(E_i) \mathbb{E}(U | E_j) \mathbb{P}(E_j) . \end{aligned}$$

In our case, we want to compute $\text{Var}(q_{n+1}(\mathbf{x}_{n+1}) | \mathcal{A}_n) := \text{Var}_n(q_{n+1}(\mathbf{x}_{n+1}))$. Since the events $\{Z \in B_i\}_{1 \leq i \leq L}$ are mutually exclusive and exhaustive, we can apply Lemma 5.1:

$$\begin{aligned} \text{Var}_n(q_{n+1}(\mathbf{x}_{n+1})) &= \sum_{i=1}^L \text{Var}_n(m_{n+1}(\mathbf{x}_{n+1}^q(B_i)) | Z \in B_i) \mathbb{P}_n(Z \in B_i) \\ &\quad + \sum_{i=1}^L \mathbb{E}_n(m_{n+1}(\mathbf{x}_{n+1}^q(B_i)) | Z \in B_i)^2 (1 - \mathbb{P}_n(Z \in B_i)) \mathbb{P}_n(Z \in B_i) \\ &\quad - 2 \sum_{i=2}^L \sum_{j=1}^{i-1} \mathbb{E}_n(m_{n+1}(\mathbf{x}_{n+1}^q(B_i)) | Z \in B_i) \mathbb{P}_n(Z \in B_i) \\ &\quad \times \mathbb{E}_n(m_{n+1}(\mathbf{x}_{n+1}^q(B_j)) | Z \in B_j) \mathbb{P}_n(Z \in B_j) . \end{aligned}$$

Thanks to equation (18), we get

$$m_{n+1}(\mathbf{x}_{n+1}^q(B_i)) = m_n(\mathbf{x}_{n+1}^q(B_i)) - k_n(\mathbf{x}_{n+1}^q(B_i), \mathbf{x}_{n+1})Z .$$

Then,

$$\begin{aligned} \text{Var}_n(q_{n+1}(\mathbf{x}_{n+1})) &= \sum_{i=1}^n k_n(\mathbf{x}_{n+1}^q(B_i), \mathbf{x}_{n+1})^2 \text{Var}_n(Z|Z \in B_i) \mathbb{P}_n(Z \in B_i) \\ &+ \sum_{i=1}^L (m_n(\mathbf{x}_{n+1}^q(B_i)) - k_n(\mathbf{x}_{n+1}^q(B_i), \mathbf{x}_{n+1}) \mathbb{E}_n(Z|Z \in B_i))^2 \\ &\times (1 - \mathbb{P}_n(Z \in B_i)) \mathbb{P}_n(Z \in B_i) \\ &- 2 \sum_{i=2}^L \sum_{j=1}^{i-1} (m_n(\mathbf{x}_{n+1}^q(B_i)) - k_n(\mathbf{x}_{n+1}^q(B_j), \mathbf{x}_{n+1}) \mathbb{E}_n(Z|Z \in B_j)) \mathbb{P}_n(Z \in B_i) \\ &\times (m_n(\mathbf{x}_{n+1}^q(B_j)) - k_n(\mathbf{x}_{n+1}^q(B_j), \mathbf{x}_{n+1}) \mathbb{E}_n(Z|Z \in B_j)) \mathbb{P}_n(Z \in B_j) . \end{aligned}$$

Since Z is a centered Gaussian random variable of variance $s_n(\mathbf{x}_{n+1})^{-2}$ we have:

$$P_i := \mathbb{P}_n(Z \in B_i) = \Phi(s_n(\mathbf{x}_{n+1})I_{i+1}) - \Phi(s_n(\mathbf{x}_{n+1})I_i) .$$

To conclude, we have now to find analytical forms for the quantities $\text{Var}(Z | I_i < Z < I_{i+1})$ and $\mathbb{E}(Z | I_i < Z < I_{i+1})$. To do so, let us use the following result on truncated Gaussian random variable (see [22] for proofs):

Lemma 5.2. *Let U be a real random variable such that $U \sim \mathcal{N}(\mu, \sigma)$. Let u and v be two real numbers. We have:*

$$\mathbb{E}(U|u < U < v) = \mu + \frac{\phi\left(\frac{v-\mu}{\sigma}\right) - \phi\left(\frac{u-\mu}{\sigma}\right)}{\Phi\left(\frac{v-\mu}{\sigma}\right) - \Phi\left(\frac{u-\mu}{\sigma}\right)} \sigma ,$$

$$\text{Var}(U|u < U < v) = \sigma^2 \left[1 + \frac{\frac{u-\mu}{\sigma} \phi\left(\frac{u-\mu}{\sigma}\right) - \frac{v-\mu}{\sigma} \phi\left(\frac{v-\mu}{\sigma}\right)}{\Phi\left(\frac{v-\mu}{\sigma}\right) - \Phi\left(\frac{u-\mu}{\sigma}\right)} - \left(\frac{\phi\left(\frac{u-\mu}{\sigma}\right) - \phi\left(\frac{v-\mu}{\sigma}\right)}{\Phi\left(\frac{v-\mu}{\sigma}\right) - \Phi\left(\frac{u-\mu}{\sigma}\right)} \right)^2 \right] .$$

We apply Lemma 5.2 for $U = Z$, $u = I_i$ and $v = I_{i+1}$ and conclude that

$$E(s_n(\mathbf{x}_{n+1}), I_{i+1}, I_i) := E_n(Z|Z \in B_i) = \frac{\phi(s_n(\mathbf{x}_{n+1})I_i) - \phi(s_n(\mathbf{x}_{n+1})I_{i+1})}{\Phi(s_n(\mathbf{x}_{n+1})I_{i+1}) - \Phi(s_n(\mathbf{x}_{n+1})I_i)} \frac{1}{s_n(\mathbf{x}_{n+1})} ,$$

$$\begin{aligned} V(s_n(\mathbf{x}_{n+1}), I_{i+1}, I_i) &:= \text{Var}_n(Z|Z \in B_i) \\ &= \frac{1}{s_n(\mathbf{x}_{n+1})^2} \left[1 + \frac{I_i s_n(\mathbf{x}_{n+1}) \phi(s_n(\mathbf{x}_{n+1})I_i) - s_n(\mathbf{x}_{n+1}) I_{i+1} \phi(s_n(\mathbf{x}_{n+1})I_i)}{\Phi(s_n(\mathbf{x}_{n+1})I_{i+1}) - \Phi(s_n(\mathbf{x}_{n+1})I_i)} \right. \\ &\quad \left. - \left(\frac{\phi(s_n(\mathbf{x}_{n+1})I_i) - \phi(s_n(\mathbf{x}_{n+1})I_{i+1})}{\Phi(s_n(\mathbf{x}_{n+1})I_{i+1}) - \Phi(s_n(\mathbf{x}_{n+1})I_i)} \right)^2 \right] . \end{aligned}$$

6.3 Test functions

Two-dimensional Branin function:

$$(19) \quad g(\mathbf{x}) = \left(\bar{x}_2 - \frac{5.1\bar{x}_1^2}{4\pi^2} + \frac{5\bar{x}_1}{\pi} - 6 \right)^2 + \left(10 - \frac{10}{8\pi} \right) \cos(\bar{x}_1) + 10$$

with: $\bar{x}_1 = 15 \times x_1 - 5$, $\bar{x}_2 = 15 \times x_2$.

Four-dimensional Hartman function:

$$(20) \quad g(\mathbf{x}) = \frac{-1}{1.94} \left[2.58 + \sum_{i=1}^4 C_i \exp \left(- \sum_{j=1}^4 a_{ji} (x_j - p_{ji})^2 \right) \right],$$

with

$$\mathbf{C} = \begin{bmatrix} 1.0 \\ 1.2 \\ 3.0 \\ 3.2 \end{bmatrix}, \quad \mathbf{a} = \begin{bmatrix} 10.00 & 0.05 & 3.00 & 17.00 \\ 3.00 & 10.00 & 3.50 & 8.00 \\ 17.00 & 17.00 & 1.70 & 0.05 \\ 3.50 & 0.10 & 10.00 & 10.00 \end{bmatrix}, \quad \mathbf{P} = \begin{bmatrix} 0.1312 & 0.2329 & 0.2348 & 0.4047 \\ 0.1696 & 0.4135 & 0.1451 & 0.8828 \\ 0.5569 & 0.8307 & 0.3522 & 0.8732 \\ 0.0124 & 0.3736 & 0.2883 & 0.5743 \end{bmatrix}.$$

Six-dimensional Ackley function:

$$(21) \quad g(\mathbf{x}) = 20 + \exp(1) - 20 \exp \left(-0.2 \sqrt{\frac{1}{4} \sum_{i=1}^4 x_i^2} \right) - \exp \left[\frac{1}{4} \sum_{i=1}^4 \cos(2\pi x_i) \right]$$

REFERENCES

- [1] Aurélie Arnaud, Julien Bect, Mathieu Couplet, Alberto Pisanisi, and Emmanuel Vazquez. Évaluation d'un risque d'inondation fluviale par planification séquentielle d'expériences. In *42èmes Journées de Statistique*, Marseille, France, France, 2010.
- [2] 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.
- [3] Claire Cannamela, Josselin Garnier, and Bertrand Iooss. Controlled stratification for quantile estimation. *The Annals of Applied Statistics*, pages 1554–1580, 2008.
- [4] Clément Chevalier, David Ginsbourger, Julien Bect, 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), 2014.
- [5] Clément Chevalier, David Ginsbourger, and Xavier Emery. Corrected kriging update formulae for batch-sequential data assimilation. In *Mathematics of Planet Earth*, pages 119–122. Springer, 2014.
- [6] David and Nagaraja. *Order Statistics*. Wiley, 2003.
- [7] L.C.W. Dixon and G.P. Szegö. *Towards Global Optimisation 2*, volume 2. North Holland, 1978.
- [8] Nando D. Freitas, Masrour Zoghi, and Alex J. Smola. Exponential regret bounds for gaussian process bandits with deterministic observations. In John Langford and Joelle Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 1743–1750, New York, NY, USA, 2012. ACM.
- [9] Donald Geman and Bruno Jedynak. An active testing model for tracking roads in satellite images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(1):1–14, 1996.
- [10] Robert B Gramacy and Herbert KH Lee. Bayesian treed gaussian process models with an application to computer modeling. *Journal of the American Statistical Association*, 103(483), 2008.
- [11] Marjorie Jala, Céline Lévy-Leduc, Eric Moulines, Emmanuelle Conil, and Joe Wiart. Sequential design of computer experiments for parameter estimation with application to numerical dosimetry. In *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*, pages 909–913. IEEE, 2012.

- [12] Marjorie Jala, Céline Lévy-Leduc, Éric Moulines, Emmanuelle Conil, and Joe Wiart. Sequential design of computer experiments for the assessment of fetus exposure to electromagnetic fields. *Technometrics*, 58(1):30–42, 2016.
- [13] B Kenkel. *pbivnorm: Vectorized Bivariate Normal CDF*, 2012. R package version 0.5-1.
- [14] Marc C Kennedy and Anthony O’Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464, 2001.
- [15] Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.
- [16] Jeremy Oakley. Estimating percentiles of uncertain computer code outputs. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 53(1):83–93, 2004.
- [17] Victor Picheny. Multiobjective optimization using gaussian process emulators via stepwise uncertainty reduction. *Statistics and Computing*, pages 1–16, 2013.
- [18] C.E. Rasmussen and C.K.I. Williams. *Gaussian processes for machine learning*. MIT Press, 2006.
- [19] Olivier Roustant, David Ginsbourger, and Yves Deville. Dicekriging, diceoptim: Two r packages for the analysis of computer experiments by kriging-based metamodeling and optimization. *Journal of Statistical Software*, 51(1):1–55, 2012.
- [20] Jerome Sacks, William J Welch, Toby J Mitchell, and Henry P Wynn. Design and analysis of computer experiments. *Statistical science*, pages 409–423, 1989.
- [21] Michael L. Stein. *Interpolation of spatial data: some theory for kriging*. Springer Science & Business Media, 2012.
- [22] Georges M. Tallis. The moment generating function of the truncated multi-normal distribution. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 223–229, 1961.
- [23] Julien Villemonteix, Emmanuel Vazquez, and Eric Walter. An informational approach to the global optimization of expensive-to-evaluate functions. *Journal of Global Optimization*, 44(4):509–534, 2009.

TLR IN WITH THE INSTITUT DE MATHÉMATIQUES DE TOULOUSE (CNRS UMR 5219). UNIVERSITÉ PAUL SABATIER, 118 ROUTE DE NARBONNE, 31062 TOULOUSE, FRANCE. VP IS WITH MIAT, UNIVERSITÉ DE TOULOUSE, INRA, CASTANET-TOLOSAN, FRANCE