UNCERTAINTY QUANTIFICATION FOR BAYESIAN OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Bayesian optimization is a class of global optimization techniques. In Bayesian optimization, the underlying objective function is modeled as a realization of a Gaussian process. Although the Gaussian process assumption implies a random distribution of the Bayesian optimization outputs, quantification of this uncertainty is rarely studied in the literature. In this work, we propose a novel approach to assess the output uncertainty of Bayesian optimization algorithms, which proceeds by constructing confidence regions of the maximum point (or value) of the objective function. These regions can be computed efficiently, and their confidence levels are guaranteed by the uniform error bounds for sequential Gaussian process regression newly developed in the present work. Our theory provides a unified uncertainty quantification framework for all existing sequential sampling policies and stopping criteria.

1 INTRODUCTION

The empirical and data-driven nature of data science field makes uncertainty quantification one of the central questions that need to be addressed in order to guide and safeguard decision makings. In this work, we focus on Bayesian optimization, which is effective in solving global optimization problems for complex blackbox functions. Our objective is to quantify the uncertainty of Bayesian optimization outputs. Such uncertainty comes from the Gaussian process prior, random input and stopping time. Closed-form solution of the output uncertainty is usually intractable because of the complicated sampling scheme and stopping criterion.

1.1 PROBLEM OF INTEREST AND OUR RESULTS

Let f be an underlying *deterministic* continuous function over Ω, a compact subset of $\mathbb{R}^p$. The goal of global optimization is to find the maximum of f, denoted by $\max_{x\in\Omega} f(x)$, or the point x_{max} which satisfies $f(x_{max}) = \max_{x\in\Omega} f(x)$. In many scenarios, objective functions can be expensive to evaluate. For example, f defined by a complex computer model may take a long time to run. Bayesian optimization is a powerful technique to deal with this type of problems, and has been widely used in areas including designing engineering systems (Forrester et al., 2008; Jones et al., 1998; Mockus et al., 1978), materials and drug design (Frazier & Wang, 2016; Negoescu et al., 2011; Solomou et al., 2018), chemistry (Häse et al., 2018), deep neural networks (Diaz et al., 2017; Klein et al., 2017), and reinforcement learning (Marco et al., 2017; Wilson et al., 2014).

In Bayesian optimization, f is treated as a realization of a stochastic process, denoted by Z. Usually, people assume that Z is a Gaussian process. Every Bayesian optimization algorithm defines a sequential sampling procedure, which successively generates new input points, based on the acquired function evaluations over all previous input points. Usually, the next input point is determined by maximizing an acquisition function. Examples of acquisition functions include probability of improvement (Kushner, 1964), expected improvement (Huang et al., 2006; Jones et al., 1998; Mockus et al., 1978; Picheny et al., 2013), Gaussian process upper confidence bound (Azimi et al., 2010; Contal et al., 2013; Desautels et al., 2014; Srinivas et al., 2010), predictive entropy search (Hernández-Lobato et al., 2014), entropy search portfolio (Shahriari et al., 2014), knowledge gradient (Scott et al., 2011; Wu & Frazier, 2016; Wu et al., 2017), etc. We refer to Frazier (2018); Shahriari et al. (2016) for an introduction to popular Bayesian optimization methods.

Although Bayesian optimization has received considerable attention and numerous techniques have emerged in recent years, how to quantify the uncertainty of the outputs from a Bayesian optimization algorithm is rarely discussed in the literature. Since we assume that f is a random realization of Z, x_{max} and $f(x_{max})$ should also be random. However, the highly nontrivial distributions of x_{max} and $f(x_{max})$ make uncertainty quantification rather challenging. Monte Carlo approaches can be employed to compute the posterior distributions of x_{max} and $f(x_{max})$, but they are usually computationally expensive, especially when a large number of observations are available.

Our results. We develop efficient methods to construct confidence regions of x_{max} and $f(x_{max})$ for Bayesian optimization algorithms, where function f is a realization of Gaussian process Z. Our uncertainty quantification method *does not* rely on the specific formulae or strategies, and can be applied to all existing methods in an abstract sense. We show that by using the collected data of any instance algorithm of Bayesian optimization, Algorithm 2 gives a confidence upper limit with theoretical guarantees of their confidence level in Corollary 3. To the best of our knowledge, this is the *first* theoretical result of the uncertainty quantification on the maximum estimator for Bayesian optimization. Compared with the traditional point-wise predictive standard deviation of Gaussian process regression, denoted by $\sigma(x)$, our bound is only inflated by a factor proportional to $\sqrt{\log(e\sigma/\sigma(x))}$, where σ is the prior standard deviation.

It is worth noting that uncertainty quantification typically differs from convergence analysis of algorithms. In Bayesian optimization, the latter topic has been studied more often. See, for instance, Bect et al. (2019); Calvin (2005; 1997); Ryzhov (2016); Vazquez & Bect (2010); Yarotsky (2013). These analyses do not directly lead to techniques for uncertainty quantification. Recall that in this work, we assume that the underlying function f is a realization of a Gaussian process, and therefore, the sample path properties of f, such as the smoothness, should be governed by the covariance function of the Gaussian process. This Gaussian process assumption differs from those in some existing works, e.g., Bull (2011); Astudillo & Frazier (2019); Yarotsky (2013), where the underlying function f is assumed to be a deterministic function satisfying pre-specified smoothness conditions.

2 PRELIMINARIES

In this section, we provide a brief introduction to Gaussian process regression and review some existing methods in Bayesian optimization.

2.1 GAUSSIAN PROCESS REGRESSION

Recall that in Bayesian optimization, the objective function f is assumed to be a realization of a Gaussian process Z. In this work, we suppose that Z is stationary and has mean zero, variance σ^2 and correlation function Ψ, i.e., $\mathrm{Cov}(Z(x), Z(x')) = \sigma^2\Psi(x - x')$ with $\Psi(0) = 1$. Under certain regularity conditions, Bochner's theorem (Wendland, 2004) suggests that the Fourier transform (with a specific choice of the constant factor) of Ψ, denoted by $\tilde{\Psi}$, is a probability density function and satisfies the inversion formula $\Psi(x) = \int_{\mathbb{R}^p} \cos(\omega^T x)\tilde{\Psi}(\omega)d\omega$. We call $\tilde{\Psi}$ the *spectral density* of Ψ. Some popular choices of correlation functions and their spectral densities are discussed in Section 3.1. We further assume Ψ satisfies the following condition. For a vector $\omega = (\omega_1, \ldots, \omega_p)^T$, define its l_1-norm as $\|\omega\|_1 = |\omega_1| + \ldots + |\omega_p|$.

Condition 1 *The correlation function Ψ has a spectral density, denoted by $\tilde{\Psi}$, and*

$$A_0 = \int_{\mathbb{R}^p} \|\omega\|_1 \tilde{\Psi}(\omega)d\omega < +\infty. \tag{1}$$

Remark 1 *The l_1-norm in Equation 1 can be replaced by the usual Euclidean norm. However, we use the former here because they usually have explicit expressions. See Section 3.1 for details.*

Remark 2 *Condition 1 imposes a smoothness condition on the correlation function Ψ, which is equivalent to the mean squared differentiability (Stein, 1999) of the Gaussian process Z. Note that the mean squared differentiability differs from the sample path differentiability. We refer to Driscoll (1973); Steinwart (2019) for results on the relationship between the sample path smoothness of Z (thus f) and the smoothness of correlation function Ψ.*

Suppose the set of points $X = (x_1, \ldots, x_n)$ is given. Then f can be reconstructed via Gaussian process regression. Let $Y = (Z(x_1), \ldots, Z(x_n))^T$ be the vector of evaluations of the Gaussian process at the design points. The following results are well-known and can be found in Rasmussen & Williams (2006). For any untried point x, conditional on Y, $Z(x)$ follows a normal distribution. The conditional mean and variance of $Z(x)$ are

$$\mu(x) := \mathbb{E}[Z(x)|Y] = r^T(x)K^{-1}Y, \tag{2}$$

$$\sigma^2(x) := \text{Var}[Z(x)|Y] = \sigma^2(1 - r^T(x)K^{-1}r(x)), \tag{3}$$

where $r(x) = (\Psi(x - x_1), \ldots, \Psi(x - x_n))^T, K = (\Psi(x_j - x_k))_{jk}$. Since we assume that f is a realization of Z, $\mu(x)$ can serve as a reconstruction of f.

2.2 Bayesian Optimization

In Bayesian optimization, we evaluate f over a set of input points, denoted by $x_1, \ldots, x_n$. We call them the *design points*, because these points can be chosen according to our will. There are two categories of strategies to choose design points. We can choose all the design points before we evaluate f at any of them. Such a design set is call a *fixed design*. An alternative strategy is called *sequential sampling*, in which the design points are not fully determined at the beginning. Instead, points are added sequentially, guided by the information from the previous input points and the corresponding acquired function values. An instance algorithm defines a sequential sampling scheme which determines the next input point x_{n+1} by maximizing an *acquisition function* $a(x; X_n, Y_n)$, where $X_n = (x_1, \ldots, x_n)$ consists of previous input points, and $Y_n = (f(x_1), \ldots, f(x_n))^T$ consists of corresponding outputs. The acquisition function can be either deterministic or random given X_n, Y_n. A general Bayesian optimization procedure under sequential sampling scheme is shown in Algorithm 1.

Algorithm 1 Bayesian optimization (described in Shahriari et al. (2016))

1: **Input:** A Gaussian process prior of f, initial observation data X_1, Y_1.
2: **for** $n = 1, 2 \ldots,$ **do**
3: Find $x_{n+1} = \arg\max_{x \in \Omega} a(x; X_n, Y_n)$, evaluate $f(x_{n+1})$, update data and the posterior probability distribution on f.
4: **Output:** The point evaluated with the largest $f(x)$.

A number of acquisition functions are proposed in the literature, for example:

1. Expected improvement (EI) (Jones et al., 1998; Mockus et al., 1978), with the acquisition function $a_{\text{EI}}(x; X_n, Y_n) := \mathbb{E}((Z(x) - y_n^*)\mathbf{1}(Z(x) - y_n^*)|X_n, Y_n)$, where $\mathbf{1}(\cdot)$ is the indicator function, and $y_n^* = \max_{1 \le i \le n} f(x_i)$.
2. Gaussian process upper confidence bound (Srinivas et al., 2010), with the acquisition function $a_{\text{UCB}}(x; X_n, Y_n) := \mu_n(x) + \beta_n \sigma_n(x)$, where β_n is a parameter, and $\mu_n(x)$ and $\sigma_n(x)$ are posterior mean and variance of f after nth iteration, respectively.
3. Predictive entropy search (Hernández-Lobato et al., 2014), with the acquisition function $a_{\text{PES}}(x; X_n, Y_n) := f^{(n)}(x)$, where $f^{(n)}$ is an approximate simulation via spectral sampling (Lázaro-Gredilla et al., 2010; Rahimi & Recht, 2008) from $\text{GP}(0, \Psi|X_n, Y_n)$.

Among the above acquisition functions, a_{EI} and a_{UCB} are deterministic functions of (x, X_n, Y_n), whereas a_{PES} is random because it depends on a random sample from the posterior Gaussian process. We refer to Shahriari et al. (2016) for general discussions and popular methods in Bayesian optimization.

In practice, one also needs to determine when to stop Algorithm 1. Usually, decisions are made in consideration of the budget and the accuracy requirement. For instance, practitioners can stop Algorithm 1 after finishing a fixed number of iterations (Frazier, 2018) or no further significant improvement of function values can be made (Acerbi & Ji, 2017). Although stopping criteria plays no role in the analysis of the algorithms' asymptotic behaviors, they can greatly affect the output uncertainty.

3 UNCERTAINTY QUANTIFICATION FOR BAYESIAN OPTIMIZATION

In this section, we present our uncertainty quantification methodology for Bayesian optimization in Section 3.1. In Section 3.2, we provide theoretical guarantees for the proposed uncertainty quantification method.

3.1 METHODOLOGY

Although the conditional distribution of $Z(x)$ is simple as shown in Equation 2 and Equation 3, those for x_{max} and $Z(x_{max})$ are highly non-trivial because they are nonlinear functionals of Z. In this work, we construct confidence regions for the maximum points and values using a uniform error bound for Gaussian process regression, as presented in Algorithm 2. In the rest of this work, let T be the number of iterations when an instance of Algorithm 1 stops and D_Ω be the diameter of Ω. Given n, we denote

$$X_{1:n} = (x_1, \ldots, x_{m_n}), \tag{4}$$

where each x_i is corresponding to one data point and m_n is the number of sampled points after n iterations of the algorithm, and $Y_{1:n} = (f(x_1), \ldots, f(x_{m_n}))^T$. In this work, we allow $m_n \geq 1$, which means that we can sample one point or a batch of points at a time in each iteration. We will use the notion $a \vee b := \max(a, b)$.

Algorithm 2 Confidence regions for $x_{\max}$ and $f(x_{\max})$

1: **Input:** Significance parameter t, data $X_{1:T}, Y_{1:T}$ collected from an instance of Bayesian optimization algorithm.

2: For any point $x \in \Omega$, set $r(x) = (\Psi(x - x_1), \ldots, \Psi(x - x_{m_T}))^T, K = (\Psi(x_j - x_k))_{jk}$. Calculate

$$\mu_T(x) = r(x)^T K^{-1} Y_{1:T}, \tag{5}$$

$$s_T(x) = \sqrt{\sigma^2(1 - r(x)^T K^{-1} r(x))}. \tag{6}$$

3: Compute

$$\text{UPPERCL}(x, t, X_{1:T}, Y_{1:T}) = \mu_T(x) + s_T(x)\sqrt{\log(e\sigma/s_T(x))}\left(C\sqrt{p(1 \vee \log(A_0 D_\Omega))} + t\right),$$

where A_0 is as in Condition 1, and C is a universal constant.

4: Calculate

$$CR_t^{\mathbf{seq}} := \left\{x \in \Omega : \text{UPPERCL}(x, t, X_{1:T}, Y_{1:T}) \geq \max_{1 \leq i \leq m_T} f(x_i)\right\}, \tag{7}$$

$$CI_t^{\mathbf{seq}} := \left[\max_{1 \leq i \leq m_T} f(x_i), \max_{x \in \Omega} \text{UPPERCL}(x, t, X_{1:T}, Y_{1:T})\right]. \tag{8}$$

5: **Output:** The confidence region $CR_t^{\mathbf{seq}}$ for x_{max} and the confidence interval $CI_t^{\mathbf{seq}}$ for $f(x_{max})$.

In Section3.2, we will show that under the condition that f is a realization of Z, $CR_t^{\mathbf{seq}}$ and $CI_t^{\mathbf{seq}}$ are confidence regions of x_{max} and $f(x_{max})$, respectively, with a simultaneous confidence level at least $1 - e^{-t^2/2}$, respectively. In particular, to obtain a 95% confidence region, we use $t = 2.448$.

Calculating A_0. For an arbitrary Ψ, calculation of A_0 in Equation 1 can be challenging. Fortunately, for two most popular correlation functions in one dimension, namely the Gaussian and the Matérn correlation functions (Rasmussen & Williams, 2006; Santner et al., 2003), A_0 can be calculated in closed form. The results are summarized in Table 1.

For multi-dimensional problems, a common practice is to use *product* correlation functions. Specifically, suppose $\Psi_1, \ldots, \Psi_p$ are one-dimensional correlation functions. Then their product $\Psi(x) = \prod_{i=1}^p \Psi(x_i)$ forms a p-dimensional correlation function, where $x = (x_1, \ldots, x_p)^T$. If a product

Table 1: Gaussian and Matérn correlation families, where $\Gamma(\cdot)$ is the Gamma function and $K_\nu(\cdot)$ is the modified Bessel function of the second kind.

Correlation family	Gaussian	Matérn
Correlation function	$\exp\{-(x/\theta)^2\}$	$\frac{1}{\Gamma(\nu)2^{\nu-1}}\left(\frac{2\sqrt{\nu}\lvert x\rvert}{\theta}\right)^{\nu} K_\nu\left(\frac{2\sqrt{\nu}\lvert x\rvert}{\theta}\right)$
Spectral density	$\frac{\theta}{2\sqrt{\pi}}\exp\{-\omega^2\theta^2/4\}$	$\frac{\Gamma(\nu+1/2)}{\Gamma(\nu)\sqrt{\pi}}\left(\frac{4\nu}{\theta^2}\right)^{\nu}\left(\omega^2+\frac{4\nu}{\theta^2}\right)^{-(\nu+1/2)}$
A_0	$\frac{2}{\sqrt{\pi}\theta}$	$\frac{4\sqrt{\nu}\Gamma(\nu+1/2)}{\sqrt{\pi}(2\nu-1)\theta\Gamma(\nu)}$ for $\nu > 1/2$

correlation function is used, the calculation of A_0 is easy. It follows from the elementary properties of Fourier transform that $\tilde{\Psi}(x) = \prod_{i=1}^p \tilde{\Psi}_i(x_i)$. Let X_i be a random variable with probability density function Ψ_i. Then $A_0 = \sum_{i=1}^p \mathbb{E}|X_i|$, i.e., the value of A_0 corresponding to a product correlation function is the sum of those given by the marginal correlation functions. If each Ψ_i is either a Gaussian or Matérn correlation function, then $\mathbb{E}|X_i|$'s can be read from Table 1.

Calibrating C via simulation studies. To use Equation 7 and Equation 8, we need to specify the constant C. In this work, we identify C by numerical simulations. The details are presented in Appendix F. Here we outline the main conclusions of our simulation studies.

Our main conclusions are: 1) $C = 1$ is a robust choice for most of the cases; 2) for the cases with Gaussian correlation functions or small $A_0 D_\Omega$, choosing $C = 1$ may lead to very conservative confidence regions. We suggest practitioners first consider $C = 1$ to obtain robust confidence regions. When users believe that this robust confidence region is too conservative, they can use the value in Table 2 or 3 corresponding to their specific setting, or run similar numerical studies as in Appendix F to calibrate their own C.

3.2 THEORY

To facilitate our mathematical analysis, we first state the general Bayesian optimization framework in a rigorous manner. Recall that we assume that f is a realization of a Gaussian process Z with correlation function Ψ. From this Bayesian point of view, we shall not differentiate f and Z in this section.

Denote the vectors of input and output points in the nth iteration as X_n and Y_n, respectively. Let $X_{1:n}$ and $Y_{1:n}$ be as in Section 3.1. Then, we can write the data we obtain after the nth iteration as $I_n = (X_{1:n}, Y_{1:n}^T)$. Because $X_{1:n}$ and $Y_{1:n}$ are random, the data I_n is associated with the σ-algebra $\mathcal{F}_n$, defined as the σ-algebra generated by $(X_{1:n}, Y_{1:n}^T)$. When the algorithm just starts, no data is gain and we set $I_0 = \varnothing$. The empty I_0 is associated with the trivial σ-algebra $\mathcal{F}_0$, which consists of only the empty set and the entire probability space. In each sampling-evaluation iteration, a sequential sampling strategy, which determines the next sample point or a batch of points based on the current data, is applied. This strategy can be deterministic or random, and may vary at different stages of the process. For example, one can choose initial designs and subsequent sampling points with different strategies. Clearly, such strategy should not depend on unobserved data. After each sampling-evaluation iteration, a stopping criterion is checked and to determine whether to terminate the algorithm. A stopping decision should depend only on the current data and/or prespecified values such as computational budget, and should not depend on unobserved data either. Let T be the number of iterations when the algorithm stops. Then a Bayesian optimization algorithm must satisfy the following conditions.

1. Conditional on $\mathcal{F}_{n-1}$, X_n and Z are mutually independent for $n = 1, 2, \ldots$.
2. T is a stopping time with respect to the filtration $\{\mathcal{F}_n\}_{n=0}^\infty$. We further require $1 \le T < +\infty, a.s.$, to ensure a meaningful Bayesian optimization procedure.

We shall establish a generic theory that bounds the uniform prediction error, which can be applied to any instance algorithms of Bayesian optimization. It is worth noting that several literature, including Sniekers & van der Vaart (2015); Yoo et al. (2016); Yang et al. (2017); Kuriki et al. (2019); Azzimonti et al. (2019); Azaïs et al. (2010), investigate uncertainty quantification methods which are not

within the Bayesian optimization or sequential sampling scheme, and cannot be directly applied to quantify the uncertainties of outputs of Bayesian optimization.

3.2.1 FIXED DESIGNS

We start with a simpler case, where we choose all the input points before we evaluate f at any of them. Although sequential samplings are more popular in Bayesian optimization, the fixed designs situation will serve as an important intermediate step to the general problem in Section 3.2.2. Let $X = (x_1, \ldots, x_n)$ be fixed design points, and $Y = (f(x_1), \ldots, f(x_n))^T$. The confidence region for x_{max} is then defined as

$$CR_t := \left\{ x \in \Omega : \text{UPPERCL}(x, t, X, Y) \geq \max_{1 \leq i \leq n} f(x_i) \right\}. \tag{9}$$

The confidence interval for $f(x_{max})$ is defined as

$$CI_t := \left[\max_{1 \leq i \leq n} f(x_i), \max_{x \in \Omega} \text{UPPERCL}(x, t, X, Y) \right]. \tag{10}$$

Also, we shall use the convention $0/0 = 0$ in all statements in this article related to error bounds. The following theorem states a uniform error bound for Gaussian process regression, which is the *first* theoretical result of this kind, to the best of our knowledge. The proof of Theorem 1 can be found in Appendix B.

Theorem 1 (Uncertainty quantification for fixed designs) *Suppose Condition 1 holds. Let* $M = \sup_{x \in \Omega} \frac{Z(x) - \mu(x)}{\sigma(x) \log^{1/2}(e\sigma/\sigma(x))}$, *where* $\mu(x)$ *and* $\sigma(x)$ *are given in Equation 2 and Equation 3, respectively. Then the followings are true.*

1. $\mathbb{E}M \leq C_0 \sqrt{p(1 \vee \log(A_0 D_\Omega))}$, *where* C_0 *is a universal constant,* A_0 *is as in Condition 1, and* $D_\Omega = diam(\Omega)$ *is the Euclidean diameter of* Ω.
2. *For any* $t > 0$, $\mathbb{P}(M - \mathbb{E}M > t) \leq e^{-t^2/2}$.

In practice, Part 2 of Theorem 1 is hard to use directly because $\mathbb{E}M$ is difficult to calculate accurately. Instead, we can replace $\mathbb{E}M$ by its upper bound in Part 1 of Theorem 1. We state such a result in Corollary 1. Its proof is trivial.

Corollary 1 *Under the conditions and notation of Theorem 1, for any constant* C *such that* $\mathbb{E}M \leq C\sqrt{p(1 \vee \log(A_0 D_\Omega))}$, *we have*

$$\mathbb{P}(M - C\sqrt{p(1 \vee \log(A_0 D_\Omega))} > t) \leq e^{-t^2/2},$$

for any $t > 0$, *where the constants* A_0 *and* D_Ω *are the same as those in Theorem 1.*

It is worth noting that the probability in Corollary 1 is *not* a posterior probability. Therefore, the regions given by Equation 9 and Equation 10 should be regarded as frequentist confidence regions under the Gaussian process model, rather than Bayesian credible regions. Such a frequentist nature has an alternative interpretation, shown in Corollary 2. Corollary 2 simply translates Corollary 1 from the language of stochastic processes to a deterministic function approximation setting, which fits the Bayesian optimization framework better. It shows that CR_t in Equation 9 and CI_t in Equation 10 are confidence region of x_{max} and $f(x_{max})$ with confidence level $1 - e^{-t^2/2}$, respectively.

Corollary 2 *Let* $C(\Omega)$ *be the space of continuous functions on* Ω *and* $\mathbb{P}_Z$ *be the law of* Z. *Then there exists a set* $B \subset C(\Omega)$ *so that* $\mathbb{P}_Z(B) \geq 1 - e^{-t^2/2}$ *and for any* $f \in B$, *its maximum point* x_{max} *is contained in* CR_t *defined in Equation 7, and* $f(x_{max})$ *is contained in* CI_t *defined in Equation 8.*

In practice, the shape of CR_t can be highly irregular and representing the region of CR_t can be challenging. If Ω is of one or two dimensions, we can choose a fine mesh over Ω and call UPPERCL(x, t, X, Y) for each mesh grid point x. In a general situation, we suggest calling UPPERCL(x, t, X, Y) with randomly chosen x's and using the k-nearest neighbors algorithm to represent CR_t.

3.2.2 SEQUENTIAL SAMPLINGS

In Bayesian optimization, sequential samplings are more popular, because such approaches can utilize the information from the previous responses and choose new design points in the area which is more likely to contain the maximum points. Similar to Section 3.2.1, we first quantify the uncertainty of $Z(\cdot)-\mu_T(\cdot)$. Note that $Z(\cdot)-\mu_T(\cdot)$ is generally *not* a Gaussian process, because in the sequential samplings situation, the stopping time T is random. Nonetheless, an error bound similar to that in Theorem 1 is still valid. In the following theorem, we define

$$\mu_n(x) := r_n^T(x)K_n^{-1}Y_{1:n}, \tag{11}$$

$$\sigma_n^2(x) := \sigma^2(1 - r_n^T(x)K_n^{-1}r_n(x)), \tag{12}$$

where $r_n(x) = (\Psi(x - x_1), \ldots, \Psi(x - x_{m_n}))^T, K_n = (\Psi(x_j - x_k))_{jk}$.

Theorem 2 (Uncertainty quantification for sequential samplings) *Suppose Condition 1 holds. Given an instance of Bayesian optimization algorithm, let*

$$M_n = \sup_{x\in\Omega} \frac{Z(x) - \mu_n(x)}{\sigma_n(x)\log^{1/2}(e\sigma/\sigma_n(x))},$$

where $\mu_n(x)$ and $\sigma_n(x)$ are given in Equation 11 and Equation 12, respectively. Then for any $t > 0$,

$$\mathbb{P}(M_T - C\sqrt{p(1 \vee \log(A_0 D_\Omega))} > t) \le e^{-t^2/2}, \tag{13}$$

where C, A_0, D_Ω are the same as in Corollary 1.

The proof of Theorem 2 can be found in Appendix D. The probability bound Equation 13 has a major advantage: the constant C is independent of the specific Bayesian optimization algorithm, and it can be chosen the same as that for fixed designs. This suggests that when calibrating C via numerical simulations (see Section 3.1 and Appendix F), we only need to simulate for fixed-design problems, and the resulting constant C can be used for the uncertainty quantification of all past and possible future Bayesian optimization algorithms.

Analogous to Corollary 2, we can restate Theorem 2 under a deterministic setting in terms of Corollary 3. In this situation, we have to restrict ourselves to *deterministic* instances of Bayesian optimization algorithms, in the sense that the sequential sampling strategy is a deterministic map, such as the first two examples in Section 2.2.

Corollary 3 *Let $C(\Omega)$ be the space of continuous functions on Ω and $\mathbb{P}_Z$ be the law of Z. Given a deterministic instance of Bayesian optimization algorithm, there exists a set $B \subset C(\Omega)$ so that $\mathbb{P}_Z(B) \ge 1 - e^{-t^2/2}$ and for any $f \in B$, its maximum point x_{max} is contained in $CR_t^{\mathbf{seq}}$ defined in Equation 7, and $f(x_{max})$ is contained in $CI_t^{\mathbf{seq}}$ defined in Equation 8.*

4 NUMERICAL EXPERIMENTS

We compare the performance between the proposed confidence interval $CI_t^{\mathbf{seq}}$ as in Equation 8 and the naive bound of Gaussian process. The nominal confidence levels are 95% for both methods. The naive 95% confidence upper bound, denoted by CI_G, is defined as the usual pointwise upper bound of Gaussian process, i.e.,

$$CI_G := \left[\max_{1\le i\le m_T} f(x_i), \max_{x\in\Omega} \mu_T(x) + q_{0.05}\sigma_T(x)\right],$$

where $q_{0.05}$ is the 0.95 quantile of the standard normal distribution, $\mu_T(x)$ and $\sigma_T(x)$ are given in Equation 5 and Equation 6, respectively. As suggested in Section 3.1, we use $C_0 = 1$ and $t = 2.448$ in $CI_t^{\mathbf{seq}}$. We consider the Matérn correlation functions (see Table 1) with $\nu = 1.5, 2.5, 3.5$, and $A_0 D_\Omega = 25$.

We simulate Gaussian processes on $\Omega = [0, 1]^2$ for each ν. We use optimal Latin hypercube designs (Stocki, 2005) to generate 5 initial points. We employs a_{UCB} (defined in Section 2.2) as the acquisition function, in which the parameter β_n is chosen as suggested by Srinivas et al. (2010).

We repeat the above procedure 100 times to estimate the coverage rate by calculating the relative frequency of the event $f(x_{\max}) \in CI_t^{\mathbf{seq}}$ or $f(x_{\max}) \in CI_G$. We also consider the "optimal upper bound" in the sense that we choose a constant a_ν and the confidence upper bound

$$CI_a := \left[\max_{1 \le i \le m_T} f(x_i), \max_{x \in \Omega} \mu_T(x) + a_\nu \sigma_T(x) \right],$$

such that the relative frequency of the event $f(x_{\max}) \in CI_a$ is exactly 95%, where a_ν only depends on ν. Then we plot the coverage rate of $CI_t^{\mathbf{seq}}$ and CI_G, and the width of $CI_t^{\mathbf{seq}}$, CI_G, and CI_a under 5, 10, 15, 20, 25, 30 iterations, respectively.

The comparison results are shown in Figure 1. Figure 1 shows the coverage rates and the width of the confidence intervals under different smoothness with $\nu = 1.5, 2.5, 3.5$. From the left plot in Figure 1, we find that the coverage rate of $CI_t^{\mathbf{seq}}$ is almost 100% for all the experiments, while CI_G has a lower coverage rate no more than 82%. Thus the proposed method is conservative while the naive one is permissive. Such a result shows that using the naive method may be risky in practice. The coverage results support our theory and conclusions made in Section 3.2. As shown by the right plot in Figure 1, the widths of $CI_t^{\mathbf{seq}}$ are about five times of CI_G, and about 2-2.5 times of CI_a. The ratio decreases as the number of iterations increases. The inflation in the width of confidence intervals is the cost of gaining confidence.

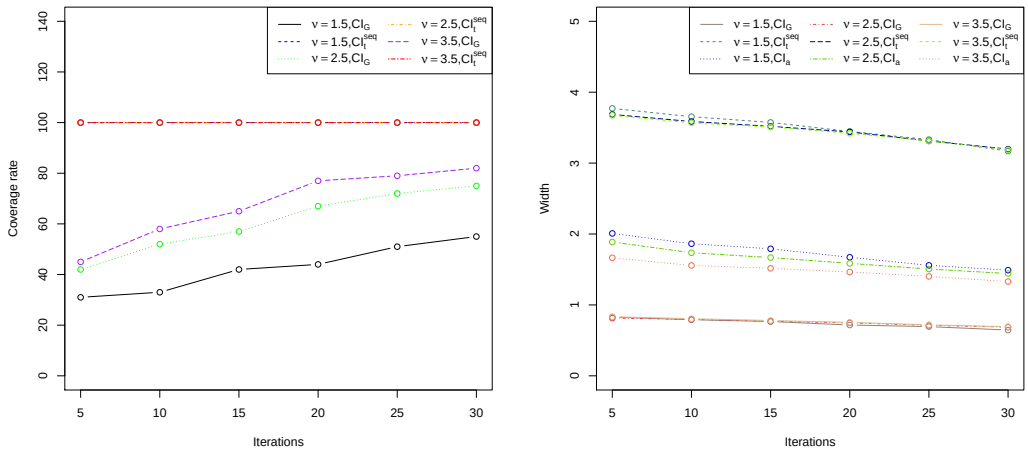

Figure 1: **Panel 1:** Coverage rates of $CI_t^{\mathbf{seq}}$ and CI_G. The nominal confidence level is 95%. **Panel 2:** Widths of $CI_t^{\mathbf{seq}}$, CI_G, and CI_a.

5 Discussion

In this work we propose a novel methodology to construct confidence regions for the outputs given by any Bayesian optimization algorithm with theoretical guarantees. To the best of our knowledge, this is the *first* result of this kind. As a cost of its high flexibility, the confidence regions may be somewhat conservative, because they are constructed based on generic probability inequalities that may not be tight enough. Nevertheless, given the fact that naive methods may be highly permissive, the proposed method can be useful when a conservative approach is preferred, such as in reliability assessments. To improve the power of the proposed method, one needs to seek for more accurate inequalities in a future work. One might also need to derive better error bounds tailored to specific acquisition functions.

REFERENCES

Luigi Acerbi and Wei Ji. Practical Bayesian optimization for model fitting with Bayesian adaptive direct search. In *Advances in Neural Information Processing Systems*, pp. 1836–1846, 2017.

Robert J Adler and Jonathan E Taylor. *Random Fields and Geometry*. Springer Science & Business Media, 2009.

Raul Astudillo and Peter I Frazier. Bayesian optimization of composite functions. *arXiv preprint arXiv:1906.01537*, 2019.

Jean-Marc Azaïs, Sophie Bercu, Jean-Claude Fort, Agnes Lagnoux, and Pierre Lé. Simultaneous confidence bands in curve prediction applied to load curves. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 59(5):889–904, 2010.

Javad Azimi, Alan Fern, and Xiaoli Z Fern. Batch Bayesian optimization via simulation matching. In *Advances in Neural Information Processing Systems*, pp. 109–117, 2010.

Dario Azzimonti, David Ginsbourger, Jérémy Rohmer, and Déborah Idier. Profile extrema for visualizing and quantifying uncertainties on excursion regions: application to coastal flooding. *Technometrics*, 61(4):474–493, 2019.

Julien Bect, François Bachoc, and David Ginsbourger. A supermartingale approach to Gaussian process based sequential design of experiments. *Bernoulli*, 2019. To appear.

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

James M Calvin. Average performance of a class of adaptive algorithms for global optimization. *The Annals of Applied Probability*, 7(3):711–730, 1997.

J.M. Calvin. One-dimensional global optimization for observations with noise. *Computers & Mathematics with Applications*, 50(1-2):157–169, 2005.

Emile Contal, David Buffoni, Alexandre Robicquet, and Nicolas Vayatis. Parallel Gaussian process optimization with upper confidence bound and pure exploration. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 225–240. Springer, 2013.

Thomas Desautels, Andreas Krause, and Joel W Burdick. Parallelizing exploration-exploitation tradeoffs in Gaussian process bandit optimization. *Journal of Machine Learning Research*, 15(1): 3873–3923, 2014.

Gonzalo I Diaz, Achille Fokoue-Nkoutche, Giacomo Nannicini, and Horst Samulowitz. An effective algorithm for hyperparameter optimization of neural networks. *IBM Journal of Research and Development*, 61(4/5):9–1, 2017.

Michael F Driscoll. The reproducing kernel Hilbert space structure of the sample paths of a Gaussian process. *Probability Theory and Related Fields*, 26(4):309–316, 1973.

Alexander Forrester, Andras Sobester, and Andy Keane. *Engineering Design via Surrogate Modelling: A Practical Guide*. John Wiley & Sons, 2008.

Peter I Frazier. A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

Peter I Frazier and Jialei Wang. Bayesian optimization for materials design. In *Information Science for Materials Discovery and Design*, pp. 45–75. 2016.

Florian Häse, Loïc M Roch, Christoph Kreisbeck, and Alán Aspuru-Guzik. Phoenics: A Bayesian optimizer for chemistry. *ACS Central Science*, 4(9):1134–1145, 2018.

José Miguel Hernández-Lobato, Matthew W Hoffman, and Zoubin Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. In *Advances in Neural Information Processing Systems*, pp. 918–926, 2014.

Deng Huang, Theodore T Allen, William I Notz, and Ning Zeng. Global optimization of stochastic black-box systems via sequential kriging meta-models. *Journal of Global Optimization*, 34(3): 441–466, 2006.

Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

Aaron Klein, Stefan Falkner, Simon Bartels, Philipp Hennig, and Frank Hutter. Fast Bayesian optimization of machine learning hyperparameters on large datasets. In *Artificial Intelligence and Statistics*, pp. 528–536, 2017.

Satoshi Kuriki, Henry P Wynn, et al. Optimal experimental design that minimizes the width of simultaneous confidence bands. *Electronic Journal of Statistics*, 13(1):1099–1134, 2019.

Harold J Kushner. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering*, 86(1):97–106, 1964.

Miguel Lázaro-Gredilla, Joaquin Quiñonero-Candela, Carl Edward Rasmussen, and Aníbal R. Figueiras-Vidal. Sparse spectrum Gaussian process regression. *Journal of Machine Learning Research*, 11(Jun):1865–1881, 2010.

Alonso Marco, Felix Berkenkamp, Philipp Hennig, Angela P Schoellig, Andreas Krause, Stefan Schaal, and Sebastian Trimpe. Virtual vs. real: Trading off simulations and physical experiments in reinforcement learning with Bayesian optimization. In *2017 IEEE International Conference on Robotics and Automation*, pp. 1557–1563, 2017.

Jonas Mockus, Vytautas Tiesis, and Antanas Zilinskas. The application of Bayesian methods for seeking the extremum. *Towards Global Optimization*, 2(117-129), 1978.

Diana M Negoescu, Peter I Frazier, and Warren B Powell. The knowledge-gradient algorithm for sequencing experiments in drug discovery. *INFORMS Journal on Computing*, 23(3):346–363, 2011.

Harald Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*, volume 63. SIAM, 1992.

Victor Picheny, David Ginsbourger, Yann Richet, and Gregory Caplin. Quantile-based optimization of noisy computer experiments with tunable precision. *Technometrics*, 55(1):2–13, 2013.

David Pollard. Empirical processes: Theory and applications. In *NSF-CBMS Regional Conference Series in Probability and Statistics*, pp. i–86. JSTOR, 1990.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, pp. 1177–1184, 2008.

C E Rasmussen and C K I Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

Ilya O Ryzhov. On the convergence rates of expected improvement methods. *Operations Research*, 64(6):1515–1528, 2016.

Thomas J Santner, Brian J Williams, and William I Notz. *The Design and Analysis of Computer Experiments*. Springer Science & Business Media, 2003.

Warren Scott, Peter Frazier, and Warren Powell. The correlated knowledge gradient for simulation optimization of continuous parameters using Gaussian process regression. *SIAM Journal on Optimization*, 21(3):996–1026, 2011.

Bobak Shahriari, Ziyu Wang, Matthew W Hoffman, Alexandre Bouchard-Côté, and Nando de Freitas. An entropy search portfolio for Bayesian optimization. *Stat*, 1050:18, 2014.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1): 148–175, 2016.

Suzanne Sniekers and Aad van der Vaart. Credible sets in the fixed design model with Brownian motion prior. *Journal of Statistical Planning and Inference*, 166:78–86, 2015.

Alexandros Solomou, Guang Zhao, Shahin Boluki, Jobin K Joy, Xiaoning Qian, Ibrahim Karaman, Raymundo Arróyave, and Dimitris C Lagoudas. Multi-objective Bayesian materials discovery: Application on the discovery of precipitation strengthened niti shape memory alloys through micromechanical modeling. *Materials & Design*, 160:810–827, 2018.

Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *the 27th International Conference on Machine Learning*, 2010.

Michael L Stein. *Interpolation of Spatial Data: Some Theory for Kriging*. Springer Science & Business Media, 1999.

Ingo Steinwart. Convergence types and rates in generic Karhunen-Loève expansions with applications to sample path properties. *Potential Analysis*, 51(3):361–395, 2019.

Rafal Stocki. A method to improve design reliability using optimal latin hypercube sampling. *Computer Assisted Mechanics and Engineering Sciences*, 12(4):393, 2005.

Aad W van der Vaart and Jon A Wellner. *Weak Convergence and Empirical Processes*. Springer, 1996.

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

Wenjia Wang, Rui Tuo, and C. F. Jeff Wu. On prediction properties of kriging: Uniform error bounds and robustness. *Journal of the American Statistical Association*, 115(530):920–930, 2020.

Holger Wendland. *Scattered Data Approximation*, volume 17. Cambridge University Press, 2004.

Aaron Wilson, Alan Fern, and Prasad Tadepalli. Using trajectory data to improve Bayesian optimization for reinforcement learning. *Journal of Machine Learning Research*, 15(1):253–282, 2014.

Jian Wu and Peter Frazier. The parallel knowledge gradient method for batch Bayesian optimization. In *Advances in Neural Information Processing Systems*, pp. 3126–3134, 2016.

Jian Wu, Matthias Poloczek, Andrew G Wilson, and Peter Frazier. Bayesian optimization with gradients. In *Advances in Neural Information Processing Systems*, pp. 5267–5278, 2017.

Yun Yang, Zuofeng Shang, and Guang Cheng. Non-asymptotic theory for nonparametric testing. *arXiv preprint arXiv:1702.01330*, 2017.

Dmitry Yarotsky. Univariate interpolation by exponential functions and Gaussian RBFs for generic sets of nodes. *Journal of Approximation Theory*, 166:163–175, 2013.

William Weimin Yoo, Subhashis Ghosal, et al. Supremum norm posterior contraction and credible sets for nonparametric multivariate regression. *The Annals of Statistics*, 44(3):1069–1102, 2016.