J Glob Optim
DOI 10.1007/s10898-013-0050-5

Optimal learning for sequential sampling with non-parametric beliefs

Emre Barut · Warren B. Powell

Received: 22 March 2012 / Accepted: 16 February 2013

Abstract We propose a sequential learning policy for ranking and selection problems, where we use a non-parametric procedure for estimating the value of a policy. Our estimation approach aggregates over a set of kernel functions in order to achieve a more consistent estimator. Each element in the kernel estimation set uses a different bandwidth to achieve better aggregation. The final estimate uses a weighting scheme with the inverse mean square errors of the kernel estimators as weights. This weighting scheme is shown to be optimal under independent kernel estimators. For choosing the measurement, we employ the knowledge gradient policy that relies on predictive distributions to calculate the optimal sampling point. Our method allows a setting where the beliefs are expected to be correlated but the correlation structure is unknown beforehand. Moreover, the proposed policy is shown to be asymptotically optimal.

Keywords Bayesian global optimization · Knowledge gradient · Non-parametric estimation

1 Introduction

We consider the problem of maximizing an unknown function over a finite set of possible alternatives. Our method can theoretically handle any number of finite alternatives but computational requirements limit this number to be on the order of thousands. We make sequential measurements from the function, obtain noisy measurements and these

This research was supported in part by grant AFOSR-FA9550-05-1-0121 from the Air Force Office of Scientific Research.

E. Barut (✉)· W. B. Powell
Department of Operations Research and Financial Engineering,
Princeton University, Princeton, NJ 08544, USA
e-mail: abarut@princeton.edu

W. B. Powell
e-mail: powell@princeton.edu

Published online: 03 March 2013

measurements are used to estimate the true values of the function. We use kernel estimation, a non-parametric estimation method and therefore we do not assume that the unknown function belongs to a certain parametric class. In addition, we do not assume Lipschitz continuity or concavity. However, we make use of the fact that if two alternatives are close to each other, their values should be similar too, a property that will arise when using continuous functions. Moreover, kernel estimation methods have convergence rates that depend on the Hölder condition number of the function. For Lipschitz functions, this condition number is equal to 1 and as the condition number increases the set of functions becomes larger. Therefore, even though no assumptions are made on the structure of the function, our estimation procedure converges faster if the function is bounded or Lipschitz. We use a Bayesian framework and start by assuming we have a normal prior distribution of beliefs about the values of the function.

This problem arises in an off-line setting, where it is known as the ranking and selection problem, and an on-line setting, where it is known as the multi armed bandit problem. Each alternative x has a reward associated with it, and we are asked to choose one from them. However, the measurements are often noisy and obtaining them could be expensive. For instance, consider a simulator for a queueing model with many inputs. Often, these simulators have very long run times and noisy results. This limits the number of different policies that can be tried in a given time, therefore finding the optimum quickly becomes a major concern as well.

Other examples of ranking and selection where a non-parametric belief model might apply include:

- *Policy optimization for energy storage.* Energy producers have to adjust the amount of energy to produce in a day to match the demand. They frequently run into the problem of over producing or underproducing energy in a day. We face the problem of tuning a parametrized policy on the basis of noisy measurements.
- *Design of fuel cells.* A fuel cell is parameterized by design parameters such as the size of the plate used for the anode or the cathode, the distance between the plates, and the concentration of the solution. These need to be tuned in a laboratory setting, requiring time and money for each experiment.
- *Simulation optimization.* The area of simulation optimization deals with optimizing functions where the function is a black box, that is, not much about the function's structure is known. Also, in most cases, evaluation from the black box take a significant amount of time, therefore a fast rate of convergence is needed.

Although the ranking and selection problem has been extensively studied, most of the previous work concentrates on problems where beliefs about the alternatives are independent [29]. Even when the measurements are used to update the global estimate, using current observations to estimate nearby alternatives (or the future benefits that might be obtained by measuring nearby points) is not often considered in the decision making process. However, whether it is the parameters for a queueing simulator or commitment levels in an energy model, the values of nearby measurements will be similar. In other words, alternatives close to each other will exhibit correlated beliefs. There is a small literature that can handle correlated beliefs; [11] makes significant use of the covariance structure for decision making, [23] fits a Gaussian process which has a fixed special correlation structure depending on the distances between the alternatives. A recent paper [37] introduces entropy minimization-based methods for Gaussian Processes. Other examples include various meta-models, where the statistical fitting procedure imposes its own covariance structure [2].

The optimization of noisy functions, broadly referred to as stochastic search, has been studied thoroughly since the seminal paper [33] which introduces the idea of stochastic gradient algorithms. An extensive coverage of the literature for stochastic search methods can be found in [35].

Optimal learning methods approach the problem in a different way and consider the value of information from each measurement. Function evaluations for optimal learning are made in a smarter way to achieve better convergence rates. There are a variety of algorithms for both discrete and continuous settings. When the alternatives are discrete, various heuristics such as interval estimation, epsilon-greedy exploration and Boltzmann exploration can be used [31,36]. The idea of making measurements based on the marginal value of information is introduced by [20] under the name $(R_1, \ldots, R_1)$ policy. This idea is extended under the name knowledge gradient using a Bayesian approach and estimates the value of measuring an alternative by the predictive distributions of the means [10]. The knowledge gradient is extended to handle correlations among the alternatives [11].

When the alternatives are continuous, commonly used methods are gradient estimation [13,35], meta-model methods such as response surface methods [2], and a series of heuristics such as tabu search and genetic algorithms [30]. Gradient estimation deals with estimating the gradient of the function in a noisy setting, and using the gradient as a direction of steepest descent. Response Surface Methodology (RSM) fits a linear regression (or a polynomial) to obtain a noisy gradient [2,8].

Recently, there is a growing trend in learning problems where the underlying process has a given structure. [6] considers a problem where they maximize over a known function whose parameters depend on an unknown monotone function. Their method is suitable for economic problems where demand or supply curves will most likely have this structure. They make use of B-splines as they are well suited to monotonicity constraints. However, their method cannot be extended to alternatives in two or more dimensions and they do not propose a well structured algorithm for their sequential measurement choices.

In the online learning setting with discrete alternatives, the optimal policy is given in [18] and [19], using a method that has become known as Gittins indices. Unfortunately, although their policy is optimal, their decision making formula requires solving for a constant dependent on the problem setting. Numerical approximations for the Gittins index are proposed in [7]. The online learning problem with continuous decisions has also been studied under various names. Agrawal has first introduced the continuum armed bandit problem and has come up with an algorithm which makes use of kernels to estimate nearby points with upper bounds on regret [1]. Tighter bounds on regret have been obtained by [26]. The response surface bandit problem, introduced in [17], considers a similar problem but assumes a polynomial structure in the rewards. They fit a quadratic surface to the rewards and use interval estimation methods. A recent paper, [34], introduces one-step ahead policies for online learning problems [25], more detail about their algorithm is given in Sect. 4.2.

We deal with an offline learning setting where the beliefs are correlated. We make use of the knowledge gradient with correlated beliefs [11]. This method, which uses a lookup table belief structure, is explained in detail in Sect. 4.1. We use a version of this knowledge gradient policy, although we implement a more sophisticated estimation procedure based on aggregation of kernels. Our approach is a general case of the method proposed by [27], where the estimators are hierarchical aggregates of the values. Our policy can also be seen as an extension of the knowledge gradient with linear beliefs [28] to non-parametric beliefs.

This paper makes the following contributions: (1) We propose a sequential Bayesian learning method that aggregates a set of estimators. (2) We construct a framework for the knowledge gradient with correlated beliefs where non-parametric estimation methods can be used. (3) We show experimentally that our method is competitive and enjoys high convergence rates.

We first introduce our model in Sect. 2. In Sect. 3, we describe our kernel estimation method, which uses a dictionary of bandwidths to circumvent the bandwidth optimization problem. In Sect. 4, we derive the knowledge gradient for this model. In Sect. 5, we present an asymptotic convergence proof. A demonstration of our algorithm is given in Sect. 6 and we propose an extension of our policy in Sect. 7. Finally in Sect. 8 we numerically compare our algorithm to other offline learning methods and present our numerical results.

2 Model

We denote the unknown function $\mu(x) : \mathcal{X} \longmapsto \mathbb{R}$, where $\mathcal{X} \subset \mathbb{R}^d$ is a finite set with M elements, in other words $\mathcal{X} = \{x_1, \ldots, x_M\}$ where $x_i \in \mathbb{R}^d$. With an abuse of notation, we also use μ_x for $\mu(x)$. We make sequential measurements from μ_x at time steps $n \in \mathbb{N}_+$. At time n, we decide to measure $\mu_{x^n} = \mu(x^n)$ and we observe

$$y_x^{n+1} = \mu_x + \varepsilon_x^{n+1},$$

where the sampling error ε_x^{n+1} is assumed to be independent from other errors and have a normal distribution with zero mean and known variance λ_x. That is, $\varepsilon_x^{n+1} \sim \mathcal{N}(0, \lambda_x)$. For the sake of simplicity, we sometimes use $\beta_x^\varepsilon = \lambda_x^{-1}$ to denote the precision of the measurement.

We let the filtration $\mathcal{F}^n$ be the sigma-algebra generated by $\{(x^0, y_{x_0}^1), \ldots, (x^{n-1}, y_{x_{n-1}}^n)\}$. As the decisions are made progressively, the decision at time n, x^n, will depend on the outcomes of the previous samples. In other words, x^n is an $\mathcal{F}^n$-measurable random variable.

We let $\mathbb{E}\left[\bullet|\mathcal{F}^n\right] = \mathbb{E}^n\left[\bullet\right]$ be the conditional expectation with respect to $\mathcal{F}^n$. We use $\mu_x^n = \mathbb{E}^n\left[\mu_x\right]$ to indicate our estimate for μ_x at time step n.

We assume that we have a Gaussian prior on the value of μ, that is,

$$\mu \sim \mathcal{N}\left(\mu^0, \boldsymbol{\Sigma}^0\right).$$

Our goal is to find the optimum point in an offline learning setting. For offline learning, we consider the case where we are allowed to make N measurements before making our final decision at time step $n = N$, when we choose

$$x^N = \arg\max_{x \in \mathcal{X}} \mu_x^N.$$

We denote by Π the set of admissible measurement policies. The problem of finding the best policy can be written as,

$$\sup_{\pi \in \Pi} \mathbb{E}^\pi \left[\max_{x \in \mathcal{X}} \mu_x^N\right],$$

where $\mathbb{E}^\pi$ denotes the expectation taken over possible outcomes when the policy $\pi \in \Pi$ is used.

For the online learning problems, we obtain the reward as we measure and alternative, therefore, the problem of finding the best policy is,

$$\sup_{\pi \in \Pi} \mathbb{E}^\pi \left[\sum_{n=0}^{N} \gamma^n \mu_{x^n}\right],$$

where γ, the discount factor is between 0 and 1 and N is the horizon of the problem. If γ is strictly smaller than 1, N can also be taken as infinity.

3 Estimation of μ_x

We propose a method that aggregates a set of different kernel estimation methods denoted by $\mathcal{K}$. By this we mean that the elements of $\mathcal{K} = \{0, 1, \ldots, i, \ldots, k\}$, use different estimation methods (Nadaraya-Watson versus higher order polynomial regression) and/or different bandwidths. This allows us to have a range of estimators that utilize different bandwidths. For any $i \in \mathcal{K}$, the estimate for μ_x at time n is denoted by $\mu_x^{i,n}$. Similarly we will use K_i to denote the kernel function used for the estimator i. We let $\mu_x^{0,n}$ be the sample mean estimate for μ_x, which may simply be the prior if there are no observations at x. Furthermore, although our method can be used with any non-parametric estimation method that uses linearly weighted sample averages (local linear estimation, Nadaraya-Watson, Gasser-Muller etc.), for the sake of simplicity and ease of presentation we work with the Nadaraya-Watson estimator. That is the estimate using kernel i is given by

$$\mu_x^{i,n} = \frac{\sum_{x' \in \mathcal{X}} K_i(x, x') \mu_{x'}^{0,n}}{\sum_{x' \in \mathcal{X}} K_i(x, x')}.$$

All of our results can easily be extended to other weighted estimation methods.

The main estimate for μ_x at time n is formed by taking a weighted average of these estimation methods. The weights are both iteration and state-dependent, and we denote each weight by $w_x^{i,n}$, producing the estimator

$$\mu_x^n = \sum_{i \in \mathcal{K}} w_x^{i,n} \mu_x^{i,n}.$$

Aggregating different estimates to obtain an overall estimate has been studied rigorously in both statistics and machine learning communities [5,12,24]. However, the focus is either prediction or estimation in both of these literatures. [24] proposes a stochastic gradient algorithm which is used to decrease the estimation error $\|\mu - \mu^n\|_2$. The same problem is studied in [5] where the weights are sequentially determined. Finally, the boosting algorithm uses a reweighted aggregation scheme to increase the accuracy of prediction [12].

Before introducing the weights we use, we make an assumption regarding our estimation procedures. We also note that our method can be used with any set of weights and the convergence results still hold if these weights go to zero for biased estimators.

Assumption 1 For a given kernel $i \in \mathcal{K}$, we assume the value of the random variable $\mu_x^i = \frac{\sum_{x' \in \mathcal{X}} K_i(x, x') \mu_{x'}}{\sum_{x' \in \mathcal{X}} K_i(x, x')}$ is distributed by $\mu_x^i \sim \mathcal{N}\left(\mu_x, \nu_x^i\right)$, where ν_x^i is the variance of $\left(\mu_x^i - \mu_x\right)$ under our prior belief. Furthermore, $\left(\mu_x^i - \mu_x\right)$ is distributed independently from $\left(\mu_x^{i'} - \mu_x\right)$ where $i, i' \in \mathcal{K}$ and $i \neq i'$.

Essentially, this is an assumption on $\boldsymbol{\Sigma}^0$. Because our weights for the kernels are adaptive, our assumption on $\boldsymbol{\Sigma}$ changes as we collect more measurements. The normality assumption of the kernel estimate is satisfied easily if we use an empirical Bayes approach and take $\mu_x^{0,0}$ to be constant for all x. If the prior has a different structure, then the kernel bandwidths have to be chosen such that $\mu_x^{i,0} = \mu_x^{0,0}$ for all $i \in \mathcal{K}$. This is easily doable by solving a linear system of equations. The independence assumption requires that for each point, the kernels cover mutually disjoint intervals. In other words, this assumption requires that kernels with larger bandwidths do not make use of the measurements closer to the center of the kernel. However, in our numerical experiments, we did not find any difference in the empirical performance of our method when we used such kernels instead of kernels with overlapping domains.

Note that the values μ_x^i are random variables that depend on μ_x. The variance of the random variable is denoted by ν_x^i. When we implement our estimation procedure, ν_x^i will be used to denote the squared bias of the ith estimator. Thus, estimators with high biases, which generally tend to be estimators with larger bandwidhts, are allowed to have their "true" value, given by μ_x^i, farther away from the true value of the function at that point. Similarly, estimators with low biases will have lower values for ν_x^i, and μ_x^i will be expected to be closer to μ_x.

Furthermore, as it will be shown in Sect. 5, our policy measures all of the alternatives infinitely often even if this assumption does not hold. Also, with this weighting scheme, the bandwidth of the final estimator goes to 0. It is a very well known fact that under these conditions, the kernel estimators will recover the true values and the effect of the bias will decline as the sample size increases.

This assumption gives us weights that are inversely proportional to the estimators' mean square errors as Proposition 1 shows (the proof is given in the Appendix).

Proposition 1 *Let $\mu_x^{i,n}$ be the posterior mean of μ_x^i at time step n, and $\left(\sigma_x^{i,n}\right)^2$ its variance. Then, under Assumption 1, the posterior belief on μ_x given observations up to time n, is normally distributed with mean and precision given by,*

$$\mu_x^n = \frac{1}{\beta_x^n}\left(\beta_x^0\mu_x^0 + \sum_{i\in\mathcal{K}}((\sigma_x^{i,n})^2 + \nu_x^i)^{-1}\mu_x^{i,n}\right),$$

$$\beta_x^n = \beta_x^0 + \sum_{i\in\mathcal{K}}((\sigma_x^{i,n})^2 + \nu_x^i)^{-1}.$$

With Proposition 1, we use the weights

$$w_x^{i,n} = \frac{((\sigma_x^{i,n})^2 + \nu_x^{i,n})^{-1}}{\sum_{i'\in\mathcal{K}}((\sigma_x^{i',n})^2 + \nu_x^{i',n})^{-1}}, \tag{1}$$

where $\left(\sigma_x^{i,n}\right)^2 := Var(\mu_x^i|\mathcal{F}^n)$ and $\nu_x^{i,n} := \left(Bias(\mu_x^{i,n}|\mathcal{F}^n)\right)^2 = (\mathbb{E}^n[\mu_x^{i,n} - \mu_x])^2$.

To summarize, after weighting each of our kernel estimators $\mu_x^{i,n}$ by $w_x^{i,n}$, our estimates for μ_x at time n will be given by,

$$\begin{aligned}\mu_x^n &= \sum_{i\in\mathcal{K}} w_x^{i,n}\mu_x^{i,n}\\ &= \sum_{i\in\mathcal{K}} \frac{((\sigma_x^{i,n})^2 + \nu_x^{i,n})^{-1}\sum_{j=1}^M \beta_x^n K_i(x,x_j)\mu_{x_j}^{0,n}}{\left(\sum_{i'\in\mathcal{K}}((\sigma_x^{i',n})^2 + \nu_x^{i',n})^{-1}\right)\left(\sum_{j=1}^M \beta_x^n K_i(x,x_j)\right)}.\end{aligned}$$

3.1 Updating equations for μ_x^n

At time n, we measure x^n and observe y_x^{n+1} and use the updating equations for the normal prior with normally distributed observations. This gives us

$$\begin{aligned}\mu_x^{0,n+1} &= \left(\beta_x^n \mu_x^{0,n} + \beta_x^\varepsilon y_x^{n+1}\right)/\beta_x^{n+1},\\ \beta_x^{n+1} &= \beta_x^n + \beta_x^\varepsilon,\end{aligned}$$

where $\mu_x^{0,n}$ is used to denote the base level estimates. $\mu_x^{i,n+1}$ is not updated unless $K_i(x, x_n) > 0$. If $K_i(x, x_n) > 0$,

$$\begin{aligned}\mu_x^{i,n+1} &= \frac{\sum_{x'\in\mathcal{X}} \beta_{x'}^{n+1} K_i(x,x')(\mu_{x'}^{0,n+1})}{\sum_{x'\in\mathcal{X}} \beta_{x'}^{n+1} K_i(x,x')}\\ &= \frac{\sum_{x'\neq x_n} \beta_{x'}^{n} K_i(x,x')(\mu_{x'}^{0,n}) + K_i(x,x_n)(\beta_{x_n}^n \mu_{x_n}^{0,n} + \beta_{x_n}^\varepsilon y_{x_n}^{n+1})}{\sum_{x'\in\mathcal{X}} \beta_{x'}^{n+1} K_i(x,x')}.\end{aligned}$$

The weights are given by,

$$w_x^{i,n} = \frac{((\sigma_x^{i,n})^2 + \nu_x^{i,n})^{-1}}{\sum_{i'\in\mathcal{K}}((\sigma_x^{i',n})^2 + \nu_x^{i',n})^{-1}}.$$

Assuming independence among the estimates of different estimation methods (which is also assumed in Assumption 1), we can use

$$\begin{aligned}\left(\sigma_x^{i,n}\right)^2 = Var(\mu_x^i|\mathcal{F}^n) &= \frac{\sum_{x'\in\mathcal{X}}(\beta_{x'}^n K_i(x,x'))^2 Var(\mu_{x'}^0|\mathcal{F}^n)}{(\sum_{x'\in\mathcal{X}} \beta_{x'}^n K_i(x,x'))^2}\\ &= \frac{\sum_{x'\in\mathcal{X}} \beta_{x'}^n K_i(x,x')^2}{(\sum_{x'\in\mathcal{X}} \beta_{x'}^n K_i(x,x'))^2}.\end{aligned}$$

We further approximate the bias using

$$\nu_x^{i,n} = (\mu_x^{i,n} - \mu_x^{0,n})^2,$$

as this is the estimate for the variance of $\mu_x - \mu_x^i$.

By Proposition 1, the variance for the final estimate is given by,

$$(\sigma_2^n)^2 = \left(\sum_{i\in\mathcal{K}}((\sigma_x^{i,n})^2 + \nu_x^i)^{-1}\right)^{-1}.$$

4 Measurement decision

In this section, we first review the Knowledge Gradient with Correlated Beliefs (KGCB) which is a ranking and selection policy [11]. Our measurement decisions are made using a variation of KGCB, and we develop this in Sect. 4.2. Knowledge gradient policies are easily adapted to deal with online learning problems [34], and we review this method in Sect. 4.3.

4.1 Knowledge gradient with correlated beliefs (KGCB)

The Knowledge Gradient with Correlated Beliefs (KGCB), an extension of the $(R_1, \ldots, R_1)$ policy [20], is a myopic policy for sequential learning for correlated alternatives [11].

Let μ be the (random) values of all alternatives $x \in \mathcal{X}$. Then, by assuming we have a prior on μ such that

$$\mu \sim \mathcal{N}(\mu^0, \boldsymbol{\Sigma}^0),$$

and by denoting $S^n = (\mu^n, \boldsymbol{\Sigma}^n)$ as the knowledge state of the state at time n, the KGCB policy picks the alternative by computing the marginal value from the information obtained by measuring x. The knowledge gradient value is given by,

$$v_x^{KG,n} = \mathbb{E}\left[\max_y \mu_y^{n+1} - \max_y \mu_y^n | S^n, x^n = x\right]. \tag{2}$$

The knowledge gradient policy then chooses

$$x^n = \arg\max_x v_x^{KG,n}.$$

In other words, in a ranking and selection setting, where we are allowed to make one more measurement before we settle on a decision, KGCB selects the alternative which produces the largest expected value from a measurement. In a Bayesian setting with Gaussian priors and Gaussian measurements, the updating equations for μ^{n+1} and $\boldsymbol{\Sigma}^{n+1}$ are given by

$$\mu^{n+1}(x) = \mu^n - \frac{y^{n+1} - \mu_x^n}{\lambda_x + \Sigma_{x,x}^n} \boldsymbol{\Sigma}^n e_x,$$
$$\Sigma^{n+1}(x) = \Sigma^n - \frac{\Sigma^n e_x e_x^T \boldsymbol{\Sigma}^n}{\lambda_x + \Sigma_{x,x}^n},$$

where e_x is a column vector and is equal to zero except at the xth location where it equals 1 [14]. Then, we can rewrite the time n conditional distribution of μ^{n+1} as,

$$\mu^{n+1} = \mu^n + \tilde{\sigma}(\boldsymbol{\Sigma}^n, x^n)Z,$$

where

$$\tilde{\sigma}(\boldsymbol{\Sigma}^n, x^n) = \frac{\boldsymbol{\Sigma}^n e_x}{\sqrt{\lambda_x + \Sigma_{x,x}^n}},$$

and Z is a standard normal random variable. Here the parameter $\tilde{\sigma}(\boldsymbol{\Sigma}^n, x^n)$ represents the predictive standard deviation of μ_x^{n+1} given $\mathcal{F}^n$. Then, plugging this in to Eq. (2) we obtain,

$$v_x^{KG,n} = \mathbb{E}[\max_y(\mu_y^n + \tilde{\sigma}_y(\boldsymbol{\Sigma}^n, x^n)Z)|S^n, x_n = x] - \max_y \mu_y^n. \tag{3}$$

To compute this value, we need to integrate the value of the normal random variable over a convex function which is given as the pointwise maximum of affine functions $\mu_y^n + \tilde{\sigma}_y(\boldsymbol{\Sigma}^n, x^n)Z$. The above decision can be computed with an algorithm of complexity O(M^2 log(M)) [11].

To demonstrate the algorithm for the calculation of $v_x^{KG,n}$, we denote $a_j^n = \mu_{x_j}^n$, $b_j^n(x) = \tilde{\sigma}_{x,x_j}(\boldsymbol{\Sigma}^n, x^n)$. The algorithm first arranges the alternatives so that the slopes $b_j^n(x)$ are in increasing order, then takes out terms a_j, b_j if there is some j' such that $b_j = b_{j'}$ and $a_j > a_{j'}$. Finally, the KGCB algorithm removes alternatives that are dominated by other alternatives, that is, it drops $a_{j'}, b_{j'}$ if for all $Z \in \mathbb{R}$ there exists some j such that $j \neq j'$ and $a_{j'} + b_{j'}Z \leq a_j + b_j Z$. After the redundant alternatives are removed with this procedure, the knowledge gradient value is given by,

$$v_x^{KG,n} = \sum_{j=1,\ldots,|\mathcal{X}|-1} (b_{j+1}^n(x) - b_j^n(x)) f\left(-\left|\frac{a_{j+1}^n - a_j^n}{b_j^n(x) - b_{j+1}^n(x)}\right|\right), \tag{4}$$

where $f(z) = \phi(z) + z\Phi(z)$, and $\phi(z)$ is the normal density and $\Phi(z)$ is the normal cumulative distribution function.

4.2 Knowledge gradient with non-parametric estimation (KGNP)

In this section, we derive the knowledge gradient when we are using a non-parametric belief structure. As given in Sect. 4.1, the knowledge gradient value for alternative x can be written as

$$v_x^{KG,n} = \mathbb{E}\left[\max_y \mu_y^{n+1} - \max_y \mu_y^n | S^n, x_n = x\right].$$

In our approach, μ_y^{n+1} is given as a weighted sum of other estimators, $\mu_y^{i,n+1}$, which can be rewritten as,

$$\mu_x^{i,n+1} = \frac{\sum_{x' \neq x_n} \beta_{x'} K_i(x, x')(\mu_{x'}^{0,n})}{\sum_{x' \in \mathcal{X}} \beta_{x'}^{n+1} K_i(x, x')} + \frac{K_i(x, x_n)(\beta_{x_n}^n \mu_{x_n}^{0,n} + \beta_{x_n}^{\varepsilon} y_{x_n}^{n+1})}{\sum_{x' \in \mathcal{X}} \beta_{x'}^{n+1} K_i(x, x')}.$$

Then, letting $A_{n+1}^i(x, x_n) = \sum_{x' \in \mathcal{X}} \beta_{x'}^n K_i(x, x') + \beta_{x_n}^{\varepsilon} K_i(x, x_n)$, we can write

$$\begin{aligned}
\mu_x^{i,n+1} &= \frac{\mu_x^{i,n}(\sum_{x' \in \mathcal{X}} \beta_{x'}^n K_i(x, x')) + \mu_x^{i,n} \beta_{x_n}^{\varepsilon} K_i(x, x_n)}{A_{n+1}^i(x, x_n)} \\
&\quad + \frac{\beta_{x_n}^{\epsilon} K_i(x, x_n)}{A_{n+1}^i(x, x_n)} \left(y_{x_n}^{n+1} - \mu_x^{i,n}\right) \\
&= \mu_x^{i,n} + \frac{\beta_{x_n}^{\varepsilon} K_i(x, x_n)}{A_{n+1}^i(x, x_n)} \left(\mu_{x_n}^n - \mu_x^{i,n}\right) + \frac{\beta_{x_n}^{\varepsilon} K_i(x, x_n)}{A_{n+1}^i(x, x_n)} \left(y_{x_n}^{n+1} - \mu_{x_n}^n\right) \\
&= \mu_x^{i,n} + \frac{\beta_{x_n}^{\varepsilon} K_i(x, x_n)}{A_{n+1}^i(x, x_n)} \left(\mu_{x_n}^n - \mu_x^{i,n}\right) + \tilde{\sigma}(x, x_n, i) Z,
\end{aligned}$$

where, $Z = \left(y_{x_n}^{n+1} - \mu_{x_n}^n\right) / \sqrt{((\sigma_{x_n}^n)^2 + \lambda_{x_n})}$ is a standard normal random variable and

$$\tilde{\sigma}(x, x_n, i) = \sqrt{((\sigma_{x_n}^n)^2 + \lambda_{x_n})} \frac{\beta_{x_n}^{\varepsilon} K_i(x, x_n)}{A_{n+1}^i(x, x_n)}.$$

Given x_n is observed at time n, using the equations above we can rewrite μ_x^{n+1} as,

$$\begin{aligned}
\mu_x^{n+1} &= \sum_{i \in \mathcal{K}} w_x^{i,n+1} \mu_x^{i,n} + \sum_{i \in \mathcal{K}} w_x^{i,n+1} \frac{\beta_{x_n}^{\varepsilon} K_i(x, x_n)}{A_{n+1}^i(x, x_n)} (\mu_{x_n}^n - \mu_x^{i,n}) \\
&\quad + \sum_{i \in \mathcal{K}} w_x^{i,n+1} \tilde{\sigma}(x, x_n, i) Z \\
&= \sum_{i \in \mathcal{K}} w_x^{i,n+1} \left(1 - \frac{\beta_{x_n}^{\varepsilon} K_i(x, x_n)}{A_{n+1}^i(x, x_n)}\right) \mu_x^{i,n} + \mu_{x_n}^n \sum_{i \in \mathcal{K}} w_x^{i,n+1} \frac{\beta_{x_n}^{\varepsilon} K_i(x, x_n)}{A_{n+1}^i(x, x_n)} \\
&\quad + Z \sum_{i \in \mathcal{K}} w_x^{i,n+1} \tilde{\sigma}(x, x_n, i).
\end{aligned}$$

As the weights in the next period will change according to the outcome of the measurement, we also need to adapt our weights for the knowledge gradient calculation. Following [27], we use predictive weights which are the expected values of the weights for the next time step. These weights are given by:

$$\bar{w}_x^{i,n}(x) \propto \left(\sum_{i \in \mathcal{K}} ((\bar{\sigma}_x^{i,n})^2 + \nu_x^{i,n})^{-1} \right)^{-1},$$

where,

$$\begin{aligned}(\bar{\sigma}_x^{i,n})^2 = Var(\mu_x^{i,n+1}|\mathcal{F}^n) &= \frac{\sum_{x' \in \mathcal{X}} (\beta_{x'}^{n+1} K_i(x,x'))^2 Var(\mu_{x'}^0|\mathcal{F}^n)}{(\sum_{x' \in \mathcal{X}} \beta_{x'}^{n+1} K_i(x,x'))^2} \\ &= \frac{\sum_{x' \in \mathcal{X}} \beta_{x'}^{n+1} K_i(x,x')^2}{(\sum_{x' \in \mathcal{X}} \beta_{x'}^{n+1} K_i(x,x'))^2}.\end{aligned}$$

Combining the equations for $\mu_x^{i,n+1}$ and the predictive weights, we obtain the knowledge gradient,

$$\upsilon_x^{KG,n}(S^n) = \mathbb{E}\left[\max_{x' \in \mathcal{X}} a_{x'}^n(x) + b_{x'}^n(x) Z | S^n \right] - \max_{x' \in X} \mu_x^n,$$

where

$$a_x^n(x_n) = \sum_{i \in \mathcal{K}} w_x^{i,n+1} \left(1 - \frac{\beta_{x_n}^{\varepsilon} K_i(x,x_n)}{A_{n+1}^i(x,x_n)} \right) \mu_x^{i,n} + \mu_{x_n}^n \sum_{i \in \mathcal{K}} w_x^{i,n+1} \frac{\beta_{x_n}^{\varepsilon} K_i(x,x_n)}{A_{n+1}^i(x,x_n)}, \quad (5)$$

$$b_x^n(x_n) = \sum_{i \in \mathcal{K}} w_x^{i,n+1} \tilde{\sigma}(x, x_n, i). \quad (6)$$

This is in the same form of KGCB as in [11], but adapted for our kernel-based belief model. By applying the procedure described in Sect. 4.1, the knowledge gradient can be computed using

$$\upsilon_x^{KG,n}(S^n) = \sum_{j=1,\ldots,|\mathcal{X}|-1} (b_{j+1}^n(x) - b_j^n(x)) f\left(- \left| \frac{a_{j+1}^n - a_j^n}{b_j^n(x) - b_{j+1}^n(x)} \right| \right).$$

4.3 Knowledge gradient for online learning

The knowledge gradient can easily be adapted to online learning problems. Consider a user who is allowed to collect information for one more time-step. After the current time period, he will repeatedly choose the alternative which he believes to be the best. That is, if we are at time step n and we are allowed to make a total of N choices, our expected reward after the current experiment is given by,

$$V^n(S^n) = (N - n + 1) \max_x \mu_x^n.$$

Then, the KG value for alternative x for online learning is given by

$$\upsilon_x^{OL-KG,n} = \mu_x^n + (N - n) \upsilon_x^{KG,n},$$

where $\upsilon_x^{KG,n}$ is the knowledge gradient value for alternative x at time step n [34].

5 Convergence results

In this section we show that our policy is asymptotically optimal almost surely. That is, with probability 1 it finds the best alternative in the limit. Since our policy is also myopically optimal by construction, this lends strong theoretical support for the hope that it will work well for finite budgets.

The proof given here is based on the convergence proof in Frazier et al. [11] for kernel estimation.

Theorem 1 *If there is at least one i such that $K_i\left(x, x'\right) > 0$ for all $x, x' \in \mathcal{X}$, then in the limit, the KGNP policy measures every alternative infinitely often, almost surely.*

Proof We start by defining Ω_0 as the almost sure event for which Lemmas 1, 2, 3, 4 (in Appendix A) hold. For any $\omega \in \Omega_0$, we let $\mathcal{X}'(\omega)$ be the random set of alternatives measured infinitely often (i.o.) with the KGNP policy. Assume that there is a set $G \subset \Omega_0$, with strictly positive probability such that for all $\omega \in G$, $\mathcal{X}'(\omega) \subsetneq \mathcal{X}$. That is with positive probability, there is at least one alternative that we measure for a finite number of times. Fix any $\omega \in G$, and let N_1 be the last time we measure an alternative outside $\mathcal{X}'(\omega)$ for this particular ω.

Let $x \in \mathcal{X}'(\omega)$; we first show that $\lim_n v_x^{KG,n} = 0$. Note that $f(z) = \phi(z) + z\Phi(z)$ is an increasing function, and $b_{j+1}^n(x) - b_j^n(x) \geq 0$ by the ordering of $b_j^n(x)$ for the KGCB procedure. Then,

$$v_x^{KG,n} \leq \sum_{j=1,\ldots,|\mathcal{X}|-1} (b_{j+1}^n(x) - b_j^n(x)) f(0). \tag{7}$$

From Lemma 4, it follows that $\lim_n b_{x'}^n(x) = 0 \forall x' \in \mathcal{X}$, and for $j = 1, \ldots, |\mathcal{X}|$ $\lim_n b_j^n(x) = 0$. Letting $n \to \infty$ in the above inequality, we obtain, $\lim_n v_x^{KG,n} = 0$. In other words, the knowledge gradient value for infinitely often sampled alternatives goes to zero in the limit.

Now, for the same $\omega \in \Omega_0$, we consider $x \notin \mathcal{X}'(\omega)$, an alternative that is not measured infinitely often. We will show that $\lim_n v_x^{KG,n} > 0$ for this alternative. Let $\mathcal{I} := \{j : \liminf_n b_j^n(x) > 0\}$. From Lemma 4, we know that $\liminf_n b_x^n(x) > 0$. As at least one alternative has to be measured infinitely often in the limit, $\mathcal{X}'(\omega)$ is non empty, and by Lemma 4, there is at least one x'' such that $\lim_n b_{x''}^n(x) = 0$. Combining the last two statements, $\mathcal{I}$ and $\mathcal{I}^C$ are both nonempty. Then, there is some $N_2 < \infty$ such that, $\min_{j \in \mathcal{I}} b_j^n(x) > \max_{j' \notin \mathcal{I}} b_{j'}^n(x)$ for all $n > N_2$. For all $n > N_2$ by the monotonicity and positivity of $f(z)$, we have

$$v_x^{KG,n} \geq \min_{j \in \mathcal{I}, j' \in \mathcal{I}^C} (b_j^n(x) - b_{j'}^n(x)) f\left(-\left|\frac{a_{j+1}^n - a_j^n}{b_j^n(x) - b_{j+1}^n(x)}\right|\right).$$

Now let $U \coloneqq \sup_{n,j,x} |a_j^n(x)|$. By Lemma 2, $U < \infty$. Then, $\sup_{n,j,x} |a_j^n(x) - a_{j+1}^n(x)| \leq 2U$. And for all $n > N_2$, by monotonicity of $f(z)$, we have

$$v_x^{KG,n} \geq \min_{j \in \mathcal{I}, j' \in \mathcal{I}^C} (b_j^n(x) - b_{j'}^n(x)) f\left(-\frac{2U}{b_j^n(x) - b_{j'}^n(x)}\right).$$

Letting, $b^* := \min_{j \in \mathcal{I}} b_j^n(x) > 0$, we take the limit in n, and by the continuity of $f(z)$, we obtain

$$\lim_n v_x^{KG,n} \geq b^* f\left(\frac{-2U}{b^*}\right) > 0. \tag{8}$$

Then, for $x' \notin \mathcal{X}'$, $\lim_n \nu_{x'}^{KG,n} > 0$, and for $x \in \mathcal{X}'$, $\lim_n \nu_x^{KG,n} = 0$. For $x' \notin \mathcal{X}'$, there will be some $n > N_1$ such that $\nu_{x'}^{KG,n} > \nu_x^{KG,n} \forall x \in \mathcal{X}'$. That is, for some time after N_1, we will choose to measure an alternative outside $\mathcal{X}'$. However, this contradicts our first assumption that $\mathcal{X}'(\omega) \subsetneq \mathcal{X}$ and that there was a last time N_1 that we stopped measuring alternatives outside $\mathcal{X}'(\omega)$. Then, $\mathcal{X}'(\omega) = \mathcal{X}$ for all $\omega \in \Omega_0$, that is we measure each alternative infinitely often. □

Corollary 1 *Under the KGNP policy,* $\lim_n \mu_x^n = \mu_x$ *a.s. for each alternative* x.

Proof By Theorem 1, every x is measured infinitely often. Then by the strong law of large numbers,

$$\lim_n \mu_x^{0,n} = \mu_x \qquad (a.s.).$$

Note that as all alternatives which are sampled infinitely often, we have,

$$\lim_n (\sigma_x^{i,n})^2 \rightarrow 0,$$

for all $i \in \mathcal{K}, x \in \mathcal{X}$. Now, fix $x \in \mathcal{X}$, and $\omega \in \Omega$, and let $\mathcal{K}' = \{i \in \mathcal{K} : \lim_n \nu_x^{i,n}(\omega) = 0\}$. Following the previous statement, these are the kernels which are equal to the true value in the limit. Then, for any $i \notin \mathcal{K}'$, although $\lim_n (\sigma_x^{i,n})^2 \rightarrow 0$, as the estimator will be biased ($\lim_n \nu_x^{i,n}(\omega) \neq 0$), hence

$$\lim_n w_x^{i,n} \longrightarrow 0.$$

That gives

$$\lim_n \mu_x^n = \lim_n \sum_{i \in \mathcal{K}} w_x^{i,n} \mu_x^{i,n} = \lim_n \sum_{i' \in \mathcal{K}'} w_x^{i',n} \mu_x^{i',n} = \lim_n \mu_x^{0,n} = \mu_x.$$

□

In practice it is impossible to measure alternatives infinitely often, and it is reasonable to stop when there is a high probability that the best alternative is chosen. If Assumption 1 holds, then by invoking Proposition 1, we can use the variance of the estimator as a measure of confidence.

Corollary 2 *Let* $x^* = \arg\max_x \mu_x^n$, *and define* κ_x *as*

$$\kappa_x = \mathbb{E}\Big[\Phi\Big(\frac{\mu_{x^*} - z_x}{\sigma_{x^*}^n}\Big)\Big],$$

where $z_x \sim \mathcal{N}\big(\mu_x^n, (\sigma_x^n)^2\big)$ *with* σ_x^n *given in Proposition 1. If Assumption 1 holds and if we stop measuring when*

$$\sum_{x \neq x^*} \kappa_x \leq \delta,$$

then, x^* *is the alternative with the highest value with probability 1-*δ.

The proof is trivial and is omitted. The result follows easily by calculating the probability that a normal random variable is larger than another normal random variable, and then by bounding above the probability with a union bound. Also note that an analytical form for κ_x does not exist. It can be estimated by using Monte Carlo methods or Laplace approximation.

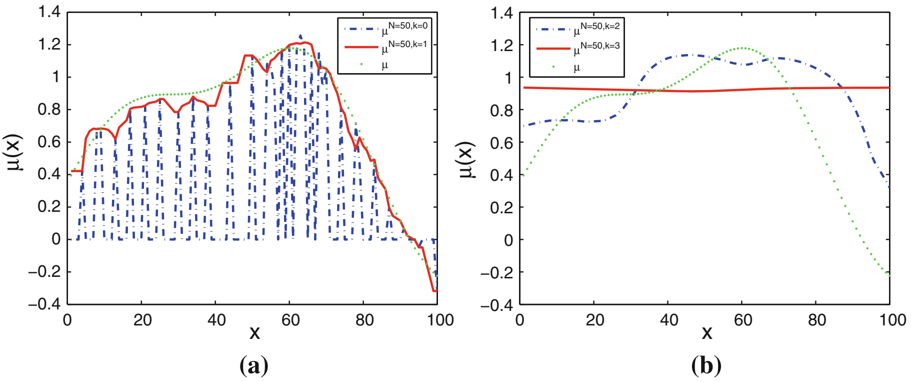


Fig. 1 Estimates given by different kernel estimation methods. On the *left* (Fig. 1a) are two estimators that use local bandwidths (h=1 in *blue* and h=4 in *red*). The true value of the function (μ_x) is shown in *green*. More global estimators (h=32 (*blue*) and h=128 (*red*)) are given on the *right* (Fig. 1b). (Color figure online)

6 KGNP demonstration

To show how our method works, we consider maximizing over a one-dimensional Gaussian process with correlation coefficient $\rho = 0.40$ and measurement variance $\lambda = 0.01$. More details about these functions are given in Sect. 8.1.1. The generated function is plotted by dotted lines in Fig. 1a, b. We start with a non-informative prior, where we pick $\mu_x^0 = 0$ and $\beta_x^0 = 0$ for all alternatives x. For bandwidths of the kernels we choose $h = \{4, 32, 128\}$ as our dictionary. Each estimation method $k_i \in \mathcal{K}$ uses a local linear fit and the kernel function is Epanechnikov with bandwidth h_i. Local linear fitting is used as it is known to have less asymptotic bias and variance than Nadaraya-Watson or Gauss-Muller estimates when the points are highly clustered [9].

We run our policy for 50 time steps, and plot the estimates at the base level (k_0) and with k_1 in Fig. 1a. In Fig. 1b, we plot our estimates with k_2 and k_3. The combined estimate which is calculated by weighting the kernel estimates by their inverse estimated MSEs is given in Fig. 2a. And in Fig. 2b, we plot the weights used for the main estimate.

7 Extension of the main algorithm

In this section, we consider an extension of the estimation method proposed in Sect. 3. This extension uses a different weighting scheme, which is common for aggregation techniques in the machine learning community. Here, we employ the sequential method proposed in [5].

The proposed method uses a tuning parameter $\eta > 0$ fixed in the beginning. Then, given that we are at time period n, we let $C_m\,(i) = \sum_{j=1}^{m} \left(y^j - \mu^i_{x^{j-1}}\right)^2$ for all $m \leq n$. Then, we choose the weights given by,

$$w_x^i = w^i = \frac{1}{n} \sum_{j=1}^{n} \frac{\exp\left(-\eta C_j\,(i)\right)}{\sum_{i' \in \mathcal{K}} \exp\left(-\eta C_j\,(i')\right)}.$$

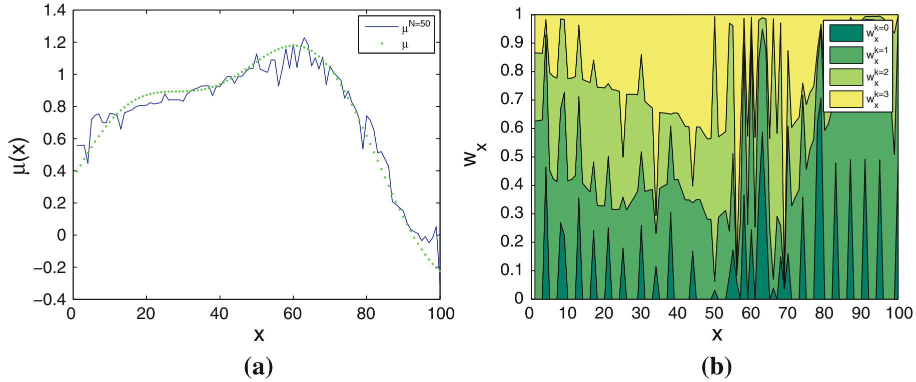


Fig. 2 Combined estimator and its weights. On the *left* (Fig. 2a) true values (μ_x) versus the combined estimator (μ_x^{50}). On the *right* (Fig. 2b): The weights used for the main estimator (w_x^{50}). The weights are inversely proportional to each estimation method's MSE. *Darker colors* represent that local estimators were used. Note that local esimation methods are used in the region around the function's maximum. (Color figure online)

To obtain their theoretical bounds on the error of this estimation procedure, Bunea and Nobel [5] pick η as

$$\eta = \left(2\,(B_1 + B_2)^2\right)^{-1},$$

where for all n and x, B_1 and B_2 satisfy,$\left|y_x^n\right| \le B_1$,$\left|\mu_x^n\right| \le B_2$ and $B_1 > B_2$. Therefore we choose to bound the highest upper value by $\max_x \left(|\mu_x| + 3\left(\beta_x^n\right)^{-1/2}\right)$ and let η as,

$$\eta = \left(2\left(\max_x \left|\mu_x^{0,n}\right| + 3\left(\beta_x^n\right)^{-1/2}\right)^2\right)^{-1}.$$

This estimator behaves very differently than the one proposed in Sect. 3 that uses MSE, and thus the resulting KGNP policy is different.

8 Numerical experiments

To evaluate our policy numerically, we ran our algorithm on continuous functions on $\mathbb{R}^d$ where the goal was finding the global maximum of the function. The functions were chosen from commonly used test functions for similar procedures. We followed an empirical Bayesian setting and started with a non-informative prior. We used zero as the prior mean and our prior precision, that is $\mu_x^0 = 0$ and $\beta_x^0 = 0$. At each time step, we evaluated the function and obtained a noisy estimate. This is in line with the methods used in simulation optimization where the optimizer sees the function as a black box and only obtains the value at given points.

As our algorithm is based on problems with a finite number of alternatives, we discretized the set of alternatives and used an equispaced grid on $\mathbb{R}^d$. Although our method is theoretically capable of handling any finite number of alternatives, computational issues limited the possible number to values on the order of 10^3.

We compared our algorithm against others in three different settings. In Sect. 8.1, we present the results from applying our policy to one-dimensional Gaussian processes and

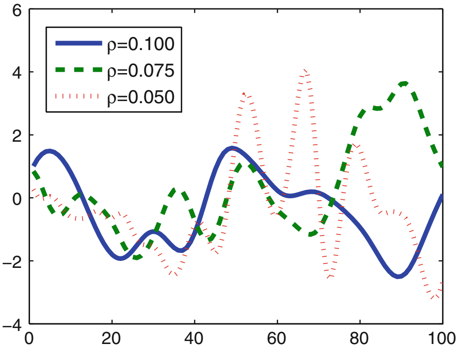


Fig. 3 Gaussian processes with different ρ

compare it against three offline learning methods. In Sect. 8.2, we use multi-dimensional test functions for comparison and in Sect. 8.3 we present an application example.

We compare our method against three alternatives: Exploration (Expl) is a policy where an alternative is chosen at random at every time step. Sequential Kriging optimization (SKO) is a black-box optimization method that fits a Gaussian process onto the observed variables [23]. Finally, the knowledge gradient with correlated beliefs (KGCB) is the method presented in Sect. 4.1. However, in our numerical comparisons, KGCB assumes that the covariance matrix is known beforehand, although this is not the case in empirical applications. Therefore, it is expected to outperform all other methods. We denote KGNP-MSE as the policy introduced in Sect. 4.2 and KGNP-EXP as the policy that uses the estimation method given in Sect. 7.

8.1 One-dimensional test functions

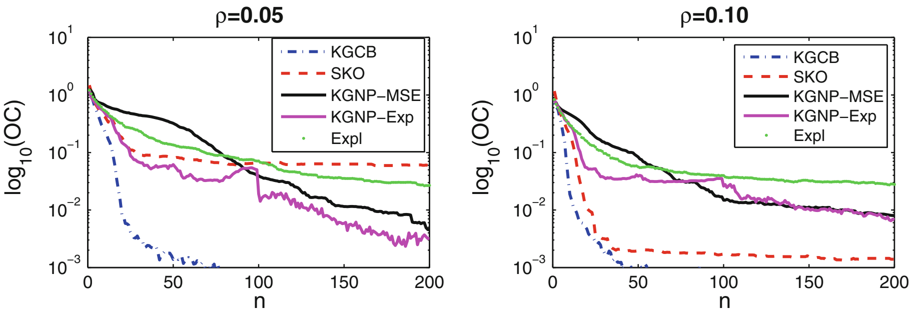
In this section, we compare our algorithm on one-dimensional Gaussian processes against three other methods listed above. Comparisons are done in two main settings: In Sect. 8.1.1, we give the results using Gaussian processes with homoscedastic covariance functions. These are multi-variate normal distributions where the covariance between two variables depends only on the distance between them. In Sect. 8.1.2, we present the results from our numerical experiments on Gaussian processes with heteroscedastic covariance functions, where the covariance terms depend both on the places of the alternatives and the distance between them.

8.1.1 Gaussian processes with homoscedastic covariance functions

In order to evaluate our method on one-dimensional functions, we generated a set of zero-mean, one-dimensional Gaussian processes on a finite interval. Our measurement set was fixed as the integers from 1 to 100 and we used the exponential covariance function

$$Cov(i, j) = \sigma^2 \exp\left(-\frac{(i-j)^2}{((M-1)\rho)^2}\right),$$

which gives a homoscedastic process with variance σ^2 and length scale ρ. A high σ^2 gives a function that varies more in the vertical axis whereas a high ρ value generates a smoother function with a smaller number of peaks and valleys. In Fig. 3, we plot randomly generated Gaussian processes with different values of ρ to show the smoothing effect as ρ is increased.



(a) $\rho = 0.05$ **(b)** $\rho = 0.10$

Fig. 4 Comparison of policies on homoscedastic GP using $\lambda = 0.01$ and various values of ρ

For all of the one-dimensional examples below, we fixed σ^2 at 2 and the measurement variance λ at 0.01. We varied ρ in each experiment. For all kernel esimators we used a Epanechnikov kernel.

We tested on two different combinations of the smoothing parameter ρ, 0.05 and 0.10. For both of these values, we generate 10 functions which gives us 30 different test functions. For each function, we tested each policy 32 times. We used opportunity cost as the performance indicator in each run:

$$\max_y \mu_y - \mu_{x^*},$$

where $x^* := \arg\max_x \mu_x^N$. We averaged the opportunity costs for policies for each different set of parameters over ρ. The only tuning parameter for our method is the set of kernel functions and the bandwidths that we start with. For these runs, we used six different kernel estimators, where each of them fit one-degree polynomials (linear fits) but with different bandwidths. We picked the bandwidth size as a geometric series ($h = 2, 2^2, \ldots, 2^6 = 64$). The opportunity costs on a log scale for different policies are given in Fig. 4.

It is seen that although the KGNP policy outperformed the exploration policy, it underperformed SKO when $\rho = 0.10$. This is expected as we are maximizing over a Gaussian process and SKO fits a Gaussian process to the evaluated function values. KGNP does not assume any structure and therefore has a slower rate of convergence. For the experiments where $\rho = 0.05$, the generated functions had more peaks and valleys, and SKO performed worse than KGNP. This is most likely due to the fact that SKO was not able to estimate ρ and therefore used a more smoothed estimator. Also, KGCB outperformed all other methods, as it was given knowledge of the true covariance stucture before it started making evaluations.

8.1.2 Gaussian processes with heteroscedastic covariance functions

Our method easily adapts to heteroscedastic covariance functions as it uses a non-parametric estimation method. To show its performance in these situations, we repeated the same experiment in the previous section using a heteroscedastic covariance function. We chose to use the Gibbs covariance function [16] as it has a similar structure with the exponential covariance function but is heteroscedastic. The Gibbs covariance function is given by,

$$Cov(i, j) = \sigma^2 \left(\frac{2l(i)l(j)}{l(i)^2 + l(j)^2} \right)^{1/2} \exp\left(-\frac{(i-j)^2}{l(i)^2 + l(j)^2} \right),$$

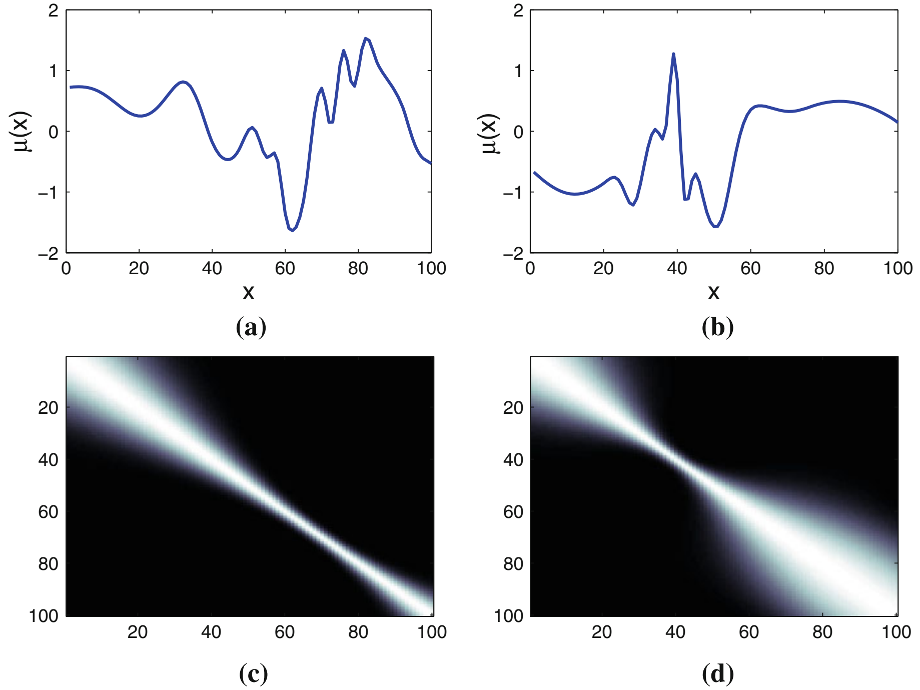


Fig. 5 Effect of varying ρ for the heteroscedastic Gibbs Gaussian process on the covariance functions and the function values: ρ values are respectively 2π and 4π. Graphs on the *top* are different functions with varying ρ values and below are their corresponding covariance matrices. *Black* and *white dots* correspond to zero and one correlation, respectively

where $l(i)$ is an arbitrary positive function in i. In our experiments, we used a horizontally shifted periodic sine curve for $l(i)$

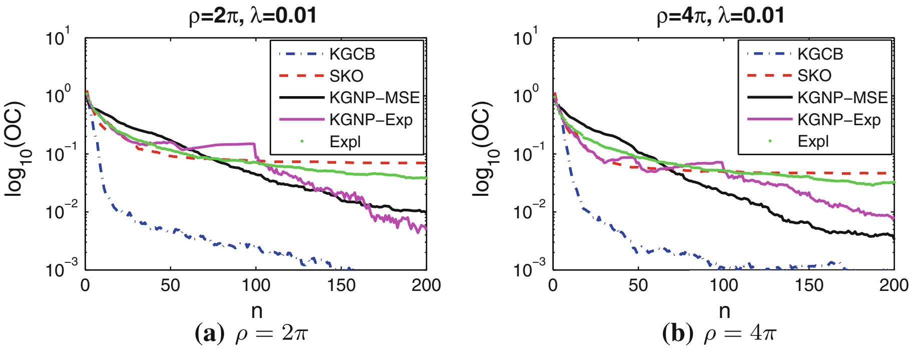
$$l(i) = 10\left(\sin\left(\rho\frac{\pi}{2}(i + c_1)\right) + 1\right) + 1,$$

where ρ determines the periodicity of the covariance function and c_1 is a random number with a uniform distribution on [0, 100] and is used to shift the curve horizontally. For the experiments, we varied ρ from 2π to 4π and the measurement variance λ in each experiment.

The effect of varying ρ for the overall covariance function and the resulting Gaussian process is given in Fig. 5.

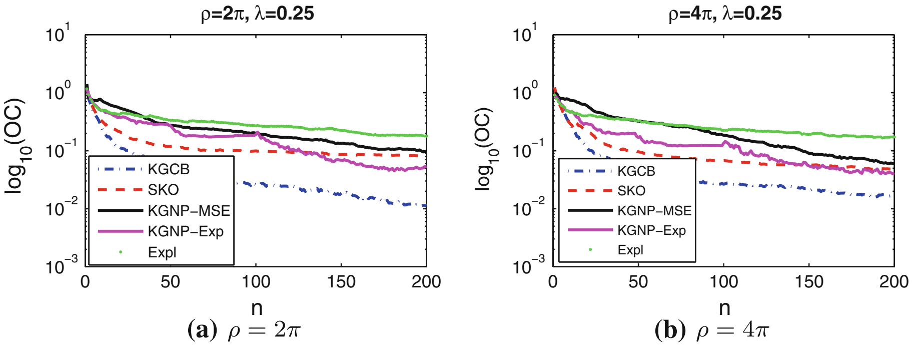
For the calculation of the opportunity cost, we followed the same setup given in the previous section. The logarithm of the opportunity costs versus iterations are given in Figs. 6 and 7.

It is seen that although SKO had a slightly faster rate of convergence in the first few iterations, it did not converge in the limit. This is due to the fact that we have a heteroscedastic covariance function and the bandwidth estimation for SKO can only handle homoscedastic Gaussian processes. One could adapt the estimation procedure in SKO to handle such covariance functions but it would require implementing non-parametric methods to estimate $l(i)$ as it can take any form. Therefore, in these setups where the function is expected to have a heteroscedastic covariance function without any specified structure, non-parametric methods will almost always have better convergence than parametric methods. Also, we



(a) $\rho = 2\pi$ **(b)** $\rho = 4\pi$

Fig. 6 Comparison of policies on heteroscedastic GP using $\lambda = 0.01$ and various values of ρ



(a) $\rho = 2\pi$ **(b)** $\rho = 4\pi$

Fig. 7 Comparison of policies on heteroscedastic GP using $\lambda = 0.25$ and various values of ρ

note that, KGCB had the perfect information of the heteroscedastic covariance function and therefore converged very rapidly.

8.2 Two-dimensional functions

We experimented with two test functions introduced in [4] and [23]. The forms, domains and the sources of these functions are given in Table 1. We compared the performance of KGNP versus SKO by testing the policies over different measurement noise levels. As KGNP works on a finite grid, we discretized each interval into 30 parts, which gives 961 (31 × 31) different alternatives. For each measurement noise level, we ran both of the policies 100 times and we did 50 iterations during each run. Opportunity cost was calculated following the same procedure in Sect. 8.1.1. To estimate the bandwidth parameter for SKO, the first six evaluations were done using a Latin hypercube square design. The results are given in Table 2.

It appears that although KGNP did not outperform SKO, the results are comparable. However, this behaviour is expected since we are using a non-parametric method that starts with almost no assumptions on the function. It is also seen that KGNP performed worse in environments with high noise, as higher observation noise with a small number of iterations forced the policy to use kernels with larger bandwidths. Therefore, using more smoothed estimates made the optimization more difficult.

Table 1 Two-dimensional functions for numerical experiments

Name	Functional form	Domain	Source
Six-hump	$f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6$	$x \in [-1.6, 2.4]$	[4]
Camelback	$+x_1x_2 - 4x_2^2 + 4x_2^4$	$\times[-.8, 1.2]$	
Tilted Branin	$f(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2$	$x \in [-5, 10]$	[23]
	$+10(1 - \frac{1}{8\pi})\cos(x_1) + 10 + \frac{1}{2}x_1$	$\times[0, 15]$	

Table 2 Expected opportunity cost after 50 iterations for two-dimensional test functions

		KGNP-MSE		KGNP-EXP		SKO	
Test function	λ	$\mathbb{E}(OC)$	SE	$\mathbb{E}(OC)$	SE	$\mathbb{E}(OC)$	SE
Six hump camelback	$.12^2$	.0310	.0012	.0504	.0062	.0321	.0030
	$.24^2$	.1243	.0281	.2365	.0249	.0495	.0044
Tilted Branin	2^2	.8414	.2661	.6815	.0650	.2390	.0158

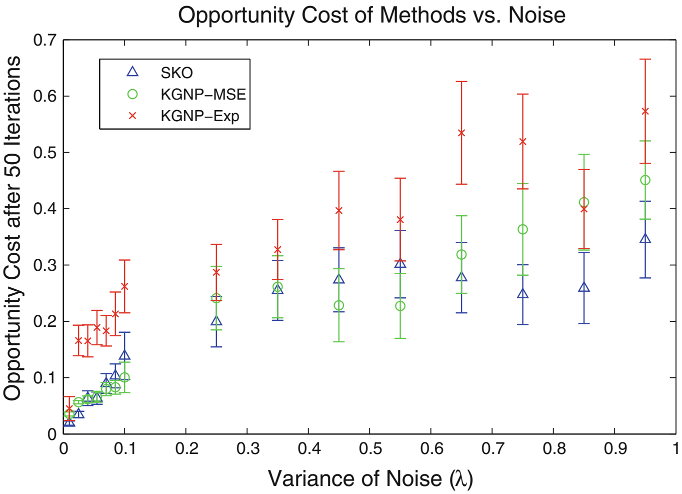


Fig. 8 Average opportunity costs for methods with respect to variance of noise (λ). *Error bars* for 95 % confidence intervals are also plotted

To illustrate the disadvantage of KGNP versus SKO in higher noise environments, we repeated the numerical experiment with the Six-Hump Camelback test function. We varied the noise variance λ from 0.01 to 1 and calculated the opportunity cost after the 50th iteration. For each noise level, we repeated the experiment for 100 times. The opportunity costs with respect to the changing noise level is given in Fig. 8.

From the results in Fig. 8, we see that SKO and KGNP-MSE perform almost at the same levels with noise variance less than 0.7. After a certain point ($\lambda = 0.75$), KGNP's performance deteriorates.

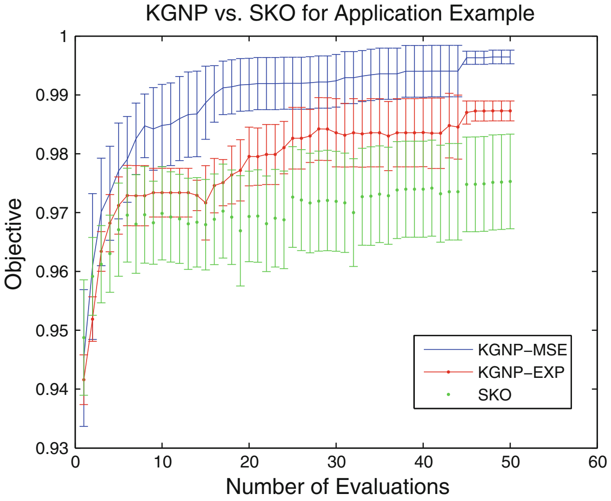


Fig. 9 Performance of KGNP and SKO for the *Black-Box* System (Objective ±2 *Standard Error*). Each policy was ran 20 times. To estimate the objective values for iteration, after each run, the values for implementation decisions are estimated using all the data

8.3 Application example

We implemented the KGNP policy to optimize a black-box simulator that estimates the value for pumped-hydro power storage. These are fairly common energy storage devices that store the energy simply by pumping the water to a higher reservoir. To release the stored energy, the water is released through turbines. Energy is stored during off-peak hours and is released during peak hours. As the price of electricity fluctuates significantly throughout the day, substantial revenues can be made if energy is stored and released at proper times.

The simulator we used had two inputs that determine the policy: The first parameter determines a price limit (for the hourly energy prices) for which all power is released from storage. The second parameter similarly defines a price limit for which we stop releasing power and start pumping in energy. In between, the level of buying decreases with exponential decay. The parameter intervals are [60, 80] and [45, 60]. Then, given two inputs within these intervals, the black-box simulated the operations of a pumped-water power storage using historical energy prices and gave an estimate of the revenue using the previously described policy.

A single evaluation from the black-box takes about a minute, and as a result we were looking for an optimization policy that can converge quickly to the optimum policy. We ran both KGNP using both weighting methods and SKO for 20 runs, each with 50 evaluations. The average of the results along with a 95 % confidence interval are given in Fig. 9.

It is seen that KGNP converged much more quickly than SKO. We also note that, since we do not know the true optimum values for this black-box function, a rigorous comparison is not possible.

9 Conclusion

In this paper, we have presented a sequential measurement policy for offline learning problems. We estimate the value function by aggregating a set of kernels with varying bandwidths. Aggregation is done using weights that are inversely proportional to the estimated mean square error. Then, we adapt the correlated knowledge gradient procedure using the covariance structure created by the kernel estimators [11]. Therefore, our method employs the knowledge gradient with a time-dependent covariance matrix where a higher weight is put on covariance matrices with better estimation.

We show that our policy is asymptotically optimal by showing it measures every alternative infinitely often and finds the best alternative in a finite set with probability 1 as the number of iterations n goes to ∞. We close with numerical results on single and two-dimensional functions. For one dimension, we test and compare our policy against several other policies on randomly generated Gaussian processes. For higher-dimensions, we employ commonly used test functions from the literature. Numerical experiments in these settings demonstrate the efficiency of our policy.

Although our policy performs very well in the numerical experiments, there is a caveat. Kernel estimation is known to suffer from the curse of dimensionality as the MSE proportional to h^d where h is the bandwidth and d is the number of dimensions. If observations lie in high dimensional spaces, non-parametric estimation is known to have a poor performance. Because of these reasons, the efficiency of our estimation method also degenerates in higher dimensions. Additive models might be used to handle this curse but this requires making more assumptions on the structure of the function.

10 Proofs

In this section, we provide the proofs for the propositions and the lemmas used in the paper. For simplicity, when there is no confusion, we use $K(x, x')$ to denote $K_i(x, x')$.

10.1 Proof of Proposition 1

Proof Let $\mathcal{C}$ be a generic subset of $\mathcal{K}$. We first show that for any such $\mathcal{C}$, the posterior of μ_x given $\mu_x^{i,n}$, for all $i \in \mathcal{C}$ is normal with mean and precision given by,

$$\mu_x^{C,n} = \frac{1}{\beta^{C,n}}\left(\beta_x^0\mu_x^0 + \sum_{i\in\mathcal{C}}((\sigma_x^{i,n})^2 + \nu_x^i)^{-1}\mu_x^{i,n}\right),$$
$$\beta_x^{C,n} = \beta_x^0 + \sum_{i\in\mathcal{C}}((\sigma_x^{i,n})^2 + \nu_x^i)^{-1}.$$

Then, the proposition follows by letting $\mathcal{C} = \mathcal{K}$.

Using induction, we first consider $\mathcal{C} = \emptyset$, then clearly the posterior is the same as the prior (μ_x^0, β_x^0) and the above equation holds as well.

Now, assume the proposed equations for the posterior distribution hold for all $\mathcal{C}$ of size m, and consider $\mathcal{C}'$ with $m + 1$ elements ($\mathcal{C}' = \mathcal{C} \cup \{j\}$). By Bayes' rule

$$\mathbb{P}_{C'}(\mu_x \in du) = \mathbb{P}_C(\mu_x \in du|Y_x^j = h) \propto \mathbb{P}_C(Y_x^j \in dh|\mu_x = u)\mathbb{P}_C(\mu_x \in du).$$

where Y_x^j stands for the observations for kernel j. Using the previous induction statement

$$\mathbb{P}_C(\mu_x \in du) = \varphi((u - \mu_x^{C,n})/\sigma_x^{C,n}).$$

By the independence assumption,

$$\begin{aligned}
&\mathbb{P}_C(Y_x^j \in dh|\mu_x = u) = \mathbb{P}(Y_x^j \in dh|\mu_x = u)\\
&= \int_{\mathbb{R}} \mathbb{P}(Y_x^j \in dh|\mu_x^k = v)\mathbb{P}(\mu_x^k = v|\mu_x = u)dv\\
&\propto \int_{\mathbb{R}} \varphi((\mu_x^{j,n} - v)/\sigma_x^{j,n})\varphi((v-u)/\sqrt{\nu_x^j})dv \propto \varphi\left(\frac{\mu_x^{j,n} - u}{\sqrt{(\sigma_x^{j,n})^2 + \nu_x^j}}\right).
\end{aligned}$$

Combining $\mathbb{P}_C(Y_x^j \in dh|\mu_x = u)$ and $\mathbb{P}_C(\mu_x \in du)$, we obtain

$$\mathbb{P}_{C'}(\mu_x \in du) \propto \varphi\left(\frac{\mu_x^{j,n} - u}{\sqrt{(\sigma_x^{j,n})^2 + \nu_x^j}}\right)\varphi((u - \mu_x^{C,n})/\sigma_x^{C,n}) \propto \varphi((u - \mu_x^{C',n})/\sigma_x^{C',n}).$$

This gives us the desired result. □

10.2 Proofs of Lemmas

This section contains the lemmas used for proving Theorem 1.

Lemma 1 *For all $x \in \mathcal{X}$, $\limsup_n \max_{m\leq n} \left|\mu_x^{0,m}\right|$ is finite almost surely (a.s.).*

Proof We fix $x \in \mathcal{X}$. For each ω, we let $N_x^n(\omega)$ the number of times we measure alternative x until time period n,

$$N_x^n(\omega) = \sum_{m\leq n-1} 1_{\{x^m = x\}}.$$

$N_x^n(\omega)$ is an increasing sequence for all ω and the limit $N_x^\infty(\omega) = \lim_{n\to\infty} N_x^n(\omega)$ exists. We bound $\left|\mu_x^{0,n}\right|$ above by,

$$\begin{aligned}
\left|\mu_x^{0,n}\right| &\leq \frac{\beta_x^0}{\beta_x^n}\left|\mu_x^{0,0}\right| + \frac{\beta_x^n - \beta_x^0}{\beta_x^n}\left|\frac{\sum_{j=1}^{n-1} 1_{\{x^i = x\}} y_x^{j+1}}{N_x^n(\omega)}\right|\\
&\leq \frac{\beta_x^0}{\beta_x^n}\left|\mu_x^{0,0}\right| + \frac{\beta_x^n - \beta_x^0}{\beta_x^n}\left|\mu_x\right| + \frac{\beta_x^n - \beta_x^0}{\beta_x^n}\left|\frac{\sum_{j=1}^{n-1} 1_{\{x^j = x\}} y_x^{j+1} - N_x^n(\omega)\mu_x}{N_x^n(\omega)}\right|\\
&= \frac{\beta_x^0}{\beta_x^n}\left|\mu_x^{0,0}\right| + \frac{\beta_x^n - \beta_x^0}{\beta_x^n}\left|\mu_x\right| + \frac{\lambda_x\left(\beta_x^n - \beta_x^0\right)}{\beta_x^n}\left|\sum_{j=1}^{n-1} 1_{\{x^j = x\}}\frac{\left(y_x^{j+1} - \mu_x\right)}{\lambda_x}\right|.
\end{aligned}$$

$\frac{\beta_x^n - \beta_x^0}{\beta_x^n}$ is bounded above by 1, and the first two terms are clearly finite, therefore we only concentrate on the finiteness of the last term. Note that $\frac{\left(y_x^{j+1} - \mu_x\right)}{\lambda_x}$ has a standard normal distribution. As the normal distribution has finite mean, we let Ω_0 be the almost sure event where $\left|y_x^j\right| \neq \infty$ for all $j \in \mathbb{N}_+$. We further divide Ω_0 into two sets,

$$\hat{\Omega}_0 = \left\{\omega \in \Omega_0 : N_x^\infty(\omega) < \infty\right\},$$

where alternative x is measured finitely many times, and

$$\hat{\Omega}_0^C = \Omega_0 \backslash \hat{\Omega}_0 = \left\{ \omega \in \Omega_0 : N_x^{\infty}(\omega) = \infty \right\}$$

where alternative x is measured infinitely often. We further define the event $\mathcal{H}_x$ as

$$\mathcal{H}_x = \left\{ \omega \in \Omega_0 : \limsup_n \max_{m \le n} \left| \mu_x^{0,m} \right| = \infty \right\}.$$

We will show that $\mathbb{P}\left(\hat{\Omega}_0 \cap \mathcal{H}_x\right) = 0$ and $\mathbb{P}\left(\hat{\Omega}_0^C \cap \mathcal{H}_x\right) = 0$ to conclude that $\mathbb{P}(\mathcal{H}_x) = \mathbb{P}\left(\hat{\Omega}_0 \cap \mathcal{H}_x\right) + \mathbb{P}\left(\hat{\Omega}_0^C \cap \mathcal{H}_x\right) = 0$.

For any $\omega \in \hat{\Omega}_0 \cap \mathcal{H}_x$, let $M_x(\omega)$ be the last time that x is measured, that is for all $n_1, n_2 \ge M_x(\omega)$, $N_x^{n_1}(\omega) = N_x^{n_2}(\omega)$. Then, we have that

$$\begin{aligned}
\sum_{j=1}^{M_x(\omega)} \lambda_x 1_{\{x^j = x\}} \left| \frac{\left(y_x^{j+1} - \mu_x\right)}{\lambda_x} \right| &= \limsup_n \max_{m \le n} \sum_{j=1}^{M_x(\omega)} \lambda_x 1_{\{x^j = x\}} \left| \frac{\left(y_x^{j+1} - \mu_x\right)}{\lambda_x} \right| \\
&= \limsup_n \max_{m \le n} \sum_{j=1}^{m} \lambda_x 1_{\{x^j = x\}} \left| \frac{\left(y_x^{j+1} - \mu_x\right)}{\lambda_x} \right| \\
&\ge \limsup_n \max_{m \le n} \left| \sum_{j=1}^{m} \lambda_x 1_{\{x^j = x\}} \frac{\left(y_x^{j+1} - \mu_x\right)}{\lambda_x} \right| \\
&\ge \limsup_n \max_{m \le n} \left| \mu_x^{0,m} \right| = \infty,
\end{aligned}$$

where $M_x(\omega) < \infty$ by construction. However, this also implies that $y_x^{j+1} = \infty$ or $y_x^{j+1} = -\infty$ for at least one i, therefore $\omega \notin \hat{\Omega}_0$ and we get a contradiction. Then, $\mathbb{P}\left(\hat{\Omega}_0 \cap \mathcal{H}_x\right) = 0$.

To show that $\mathbb{P}\left(\hat{\Omega}_0^C \cap \mathcal{H}_x\right) = 0$, we let $J_i := 1_{\{x^i = x\}} \frac{\left(y_x^{j+1} - \mu_x\right)}{\lambda_x}$ and remind that J_i has a standard normal distribution. We further define a subsequence $G(\omega) \subset \mathbb{N}_+$ by,

$$G(\omega) := \left\{ j \in \mathbb{N}_+ : 1_{\{x^j = x\}} = 1 \right\},$$

and we let $J^* := (J_i)_{i \in G(\omega)}$. By construction, $G(\omega)$ has countably infinite elements for all $\omega \in \hat{\Omega}_0^C$. Here, we make use a version of the law of iterated logarithms [3] which states that,

$$\limsup_n \max_{m \le n} \left| \bar{Z}_n \right| < \infty \; (a.s.),$$

where $\bar{Z}_n = \sum_{j=1}^n z_i / n$ and z_j are i.i.d. random variables with zero mean and variance 1. We let Ω_1 be the almost sure set where this law holds for $\bar{Z}_n = J_n^*$, and the proof follows by noting that $\mathbb{P}\left(\hat{\Omega}_0^C \cap \mathcal{H}_x \cap \Omega_1\right) = 0$. □

Lemma 2 *Assume that we have a prior on each point* $\left(\beta_x^0 > 0, \forall x \in \mathcal{X}\right)$, *then for any* x, $x' \in \mathcal{X}$, $k_i \in \mathcal{K}$, *the following are finite a.s. :* $\sup_n \left| \mu_x^{i,n} \right|$, $\sup_n \left| a_{x'}^n(x) \right|$ *and* $\sup_n \left| b_{x'}^n(x) \right|$.

Proof For any $x \in \mathcal{X}$, $k_i \in \mathcal{K}$ and $n \in \mathbb{N}$, let $p_{x'}^{i,n} = \frac{\beta_x^n K_i(x,x')}{\sum_{j=1}^M \beta_x^n K_i(x,x_j)}$. Clearly, for any $x' \in \mathcal{X}$ all $p_{x'}^{i,n} \geq 0$ and $\sum_{x' \in \mathcal{X}} p_{x'}^{i,n} = 1$. That is for any x' and n, $p_{x'}^{i,n}$ form a convex combination of $\mu_{x'}^{0,n}$. Then,

$$\sup |\mu_x^{i,n}| = \sup_n \left| \frac{\sum_{j=1}^M \beta_x^n K_i(x, x_j) \mu_{x_j}^{0,n}}{\sum_{j=1}^M \beta_x^n K_i(x, x_j)} \right| = \sup_n \left| \sum p_x^{i,n} \mu_x^{0,n} \right| \leq \sup_{n,x} |\mu_x^{0,n}|.$$

And the last term is finite by Lemma 1.

To show the finiteness of $\sup_n |a_{x'}^n(x)|$, we note that $a_{x'}^n(x)$ is a linear combination of $\mu_x^{i,n}$ and $\mu_{x'}^{i,n}$, where the weights for $\mu_x^{i,n}$ are given by $\left(1 - \frac{\beta_{x_n}^\varepsilon K(x,x_n)}{A_{n+1}^i(x,x_n)}\right)$ and the weight for $\mu_{x'}^{i,n}$ is $\sum_{i \in \mathcal{K}} w_x^{i,n+1} \frac{\beta_{x_n}^\varepsilon K(x,x_n)}{A_{n+1}^i(x,x_n)}$. These weights are between 0 and 1, and the finiteness follows.

To see $\sup_n |b_{x'}^n(x)|$, first note that for any $i \in \mathcal{K}$ and any $x, x' \in \mathcal{X}$,

$$A_{n+1}^i(x, x') = \sum_{\hat{x} \in \mathcal{X}} \beta_{\hat{x}}^n K(x, \hat{x}) + \beta_{x'}^\varepsilon K(x, x'),$$

is an increasing sequence in n. And trivially, $(\sigma_x^n)^2 = 1/\beta_x^n$ is a decreasing sequence in n. Then for any $n \in \mathbb{N}$,

$$\tilde{\sigma}(x, x', i)_n = \sqrt{((\sigma_{x'}^n)^2 + \lambda_{x'}) \frac{\beta_{x'}^\varepsilon K(x, x')}{A_n^i(x, x')}} \leq \tilde{\sigma}(x, x', i)_0 < \infty.$$

As $b_{x'}^n(x)$ is a convex combination of $\tilde{\sigma}(x, x', i)$ where the weights are given by $w_x^{i,n}$, it follows that $\sup_n |b_{x'}^n(x)|$ is finite. □

Lemma 3 *For any $\omega \in \Omega$, we let $\mathcal{X}'(\omega)$ be the random set of alternatives measured infinitely often by the KGNP policy. Fix $\omega \in \Omega$, then for any $x \notin \mathcal{X}'(\omega)$ let $x' \in \mathcal{X}$ be an alternative such that $x' \neq x$, $K_i(x, x') > 0$ for at least one $k_i \in \mathcal{K}$, and x' is measured at least once. Also assume that $\mu_x \neq \mu_{x'}$. Then, $\liminf_n \left| \mu_x^{i,n} - \mu_x^{0,n} \right| > 0$ a.s. In other words, the estimator using kernel k_i has a bias almost surely.*

Proof As $x \notin \mathcal{X}'$, there is some $N < \infty$ such that $\mu_x^{0,n} = \mu_x^{0,N}$ for all $n \geq N$. And as $\mu_x^{0,N} = \frac{\mu_x^0 + \sum_{m \leq N} \beta_x^\varepsilon y_{x_m} 1_{(x_m = x)}}{\beta_x^0 + \sum_{m \leq N} \beta_x^\varepsilon 1_{(x_m = x)}}$, it is given by a linear combination of normal random variables $\left(y_{x_m}\right)$ and is a continuous random variable.

As $x \neq x'$ is at least measured once, and $K_i(x, x') > 0$, $\mu_x^{i,n}$ contains positively weighted $\mu_{x'}^{0,n}$ terms. Also, using the assumption $\mu_{x'} \neq \mu_x$, $\mu_{x'}^{0,n}$ will not be perfectly correlated with $\mu_x^{0,n}$. Then, as both are continuous random variables, the probability that $\mu_x^{0,n}$ will be equal to any cluster point of $\mu_x^{i,n}$ is zero a.s. That is $\liminf_n \left| \mu_x^{i,n} - \mu_x^{0,n} \right| > 0$. □

Remark If μ_x are generated from a continuously distributed prior (e.g. normal distribution), then for all $x \neq x'$, $\mathbb{P}(\mu_x \neq \mu_{x'}) = 1$ and the assumption for the previous lemma holds almost surely.

Lemma 4 *For any $\omega \in \Omega$, we let $\mathcal{X}'(\omega)$ be the random set of alternatives measured infinitely often by the KGNP policy. For all $x, x' \in \mathcal{X}$, the following holds a.s.:*

- *if $x \in \mathcal{X}'$, then $\lim_n b_{x'}^n(x) = 0$ and $\lim_n b_x^n(x') = 0$,*
- *if $x \notin \mathcal{X}'$, then $\liminf_n b_x^n(x) > 0$.*

Proof We start by considering the first case, $x \in \mathcal{X}'$. If $K_i(x, x') = 0$ for all $i \in \mathcal{K}$, $b^n_{x'}(x) = b^n_x(x') = 0$ for all n by the definition. Taking $n \to \infty$ we get the result.

If $K_i(x, x') > 0$ for some $i \in K$, showing $\lim_n b^n_{x'}(x) = 0$ is equivalent to showing that for all $i \in \mathcal{K}$

$$\tilde{\sigma}(x, x', i) = \sqrt{((\sigma^n_{x'})^2 + \lambda_{x'})}\frac{\beta^\varepsilon_{x'} K(x, x')}{A^i_{n+1}(x, x')} \longrightarrow 0.$$

As noted previously, $A^i_n(x, x')$ is an increasing sequence. If $x \in \mathcal{X}'$, then we also have that, $\beta^n_x \to \infty$, and

$$\frac{1}{A^i_{n+1}(x, x')} \leq \frac{1}{\beta^n_x K(x, x')} \longrightarrow 0.$$

Therefore $\lim_n b^n_{x'}(x) = 0$ under this case as well. Showing $\lim_n b^n_x(x') = 0$, reduces to showing that,

$$\frac{1}{A^i_{n+1}(x', x)} \longrightarrow 0,$$

which is also given by above.

Now for the second result, where $K_i(x, x') > 0$ for some $i \in \mathcal{K}$ and $x \notin \mathcal{X}'$; by the definition of $b^n_x(x)$

$$b^n_x(x) \geq w^{0,n+1}_x \tilde{\sigma}(x, x, 0) = w^{0,n+1}_x \sqrt{((\sigma^n_x)^2 + \lambda_x}\frac{\beta^\varepsilon_x}{\beta^n_x + \beta^\varepsilon_x K(x, x)}.$$

For a given $\omega \in \Omega$, let N be the last time that alternative x is observed. Then, for all $n \geq N$,

$$\beta^n_x = \beta^N_x \leq \beta^0_x + N\beta^\varepsilon_x < \infty.$$

Recall that $(\sigma^n_x)^2 = 1/\beta^n_x$ and $\lambda_x = 1/\beta^\varepsilon_x$, and that these terms are finite for a finitely sampled alternative. For $\liminf_n b^n_x(x) > 0$ to hold, we only need to show that the weight stays above 0, that is,

$$\liminf_n w^{0,n}_x = \liminf_n \left(\frac{((\sigma^{0,n}_x)^2)^{-1}}{\sum_{i' \in \mathcal{K}} ((\sigma^{i',n}_x)^2 + \nu^{i',n}_x)^{-1}} \right) > 0.$$

Almost sure finiteness of the numerator has been shown above, which means we only need to show that

$$\limsup_n \sum_{i' \in \mathcal{K}} ((\sigma^{i',n}_x)^2 + \nu^{i',n}_x)^{-1} < \infty.$$

First we divide the set of kernels into two pieces. Let $\mathcal{K}_1(\omega, x)$ be the set such that, for $\omega \in \Omega$, there is at least one $x' \in \mathcal{X}'$ such that $K_i(x, x') > 0$. In other words, there is one infinitely often sampled point (x') close to our original point (x) that has influence on the prediction. Let $\mathcal{K}_2(\omega, x)$=$\mathcal{K} \backslash \mathcal{K}_1$. Now as all terms are positive,

$$\limsup_n \sum_{i' \in \mathcal{K}} ((\sigma^{i',n}_x)^2 + \nu^{i',n}_x)^{-1} \leq \limsup_n \sum_{i' \in \mathcal{K}_1} ((\sigma^{i',n}_x)^2 + \nu^{i',n}_x)^{-1}$$
$$+ \limsup_n \sum_{i' \in \mathcal{K}_2} ((\sigma^{i',n}_x)^2 + \nu^{i',n}_x)^{-1}.$$

For all $k_{i'} \in \mathcal{K}_1$, we have that by Lemma 3, $\liminf_n \nu_x^{i',n} > 0$, even if $\liminf_n (\sigma_x^{i',n})^2 = 0$, the limsup for the first term on the right are finite. Finally, for all $i' \in \mathcal{K}_2$, as none of the points using $i' \in \mathcal{K}_2$ using to predict μ_x are sampled infinitely often, letting

$$N_X = \max_{x \notin \mathcal{X}'} N_x,$$

where N_x is the last time point x is sampled, we have $N_X < \infty$. Then, β_x^n for all $x \notin \mathcal{X}'(\omega)$ is finite (and bounded above by $N_X(\max_{x \notin \mathcal{X}'} \beta_x^\varepsilon)$) and

$$\begin{aligned}
\sum_{i \in \mathcal{K}_2} ((\sigma_x^{i,n})^2 + \nu_x^{i,n})^{-1} &\leq \sum_{i \in \mathcal{K}_2} ((\sigma_x^{i,n})^2)^{-1} \\
&\leq \sum_{i \in \mathcal{K}_2} \frac{(\sum_{x' \in \mathcal{X}} \beta_{x'}^n K_i(x, x'))^2}{\sum_{x' \in \mathcal{X}} \beta_{x'}^n K_i(x, x')^2} \\
&\leq \sum_{i \in \mathcal{K}_2} \frac{(\sum_{x' \in \mathcal{X}} N_X(\max_{x \notin \mathcal{X}'} \beta_x^\varepsilon) K_i(x, x'))^2}{\sum_{x' \in \mathcal{X}} N_X(\max_{x \notin \mathcal{X}'} \beta_x^\varepsilon) K_i(x, x')^2} < \infty
\end{aligned}$$

where the last term does not contain n. Taking the limit supremum over n for both sides gives us the final result. □

References

1. Agrawal, R.: The continuum-armed bandit problem. SIAM J. Control Optim. **33**, 1926–1951 (1995)
2. Barton, R.R., Meckesheimer, M.: Chapter 18 metamodel-based simulation optimization in Simulation. In: Henderson, S.G., Nelson, B.L. (eds.). vol. 13 of Handbooks in Operations Research and Management Science. Elsevier (pp. 535–574) (2006)
3. Billingsley, P.: Probability and Measure, 3rd edn. Wiley-Interscience, New York (1995)
4. Branin, F.H.: Widely convergent method for finding multiple solutions of simultaneous nonlinear equations. IBM J. Res. Dev. **16**, 504–522 (1972)
5. Bunea, F., Nobel, A.: Sequential procedures for aggregating arbitrary estimators of a conditional mean. IEEE Trans. Inf. Theory **54**, 1725–1735 (2008)
6. Chehrazi, N., Weber, T.A.: Monotone approximation of decision problems. Oper. Res. **58**, 1158–1177 (2010)
7. Chick, S.E., Gans, N.: Economic analysis of simulation selection problems. Manag. Sci. **55**, 421–437 (2009)
8. Cochran, W.G., Cox, G.M.: Experimental Designs. Wiley, New York (1957)
9. Fan, J., Gijbels, I.: Local Polynomial Modelling and Its Applications: Monographs on Statistics and Applied Probability 66 (Chapman & Hall/CRC Monographs on Statistics & Applied Probability). Chapman & Hall, London (1996)
10. Frazier, P.I., Powell, W.B., Dayanik, S.: knowledge-gradient policy for sequential information collection. SIAM J. Control Optim. **47**, 2410–2439 (2008)
11. Frazier, P.I., Powell, W.B., Dayanik, S.: The knowledge-gradient policy for correlated normal beliefs. INFORMS J. Comput. **21**, 599–613 (2009)
12. Freund, Y., Schapire, R.: A decision-theoretic generalization of on-line learning and an application to boosting in computational learning theory. In: Vitanyi, P. (ed.) vol. 904 of Lecture Notes in Computer Science. Springer Berlin, Heidelberg (1995)
13. Fu, M.C.: Chapter 19 gradient estimation. In: Simulation. In: Henderson, S.G., Nelson, B.L. (eds.) vol. 13 of Handbooks in Operations Research and Management Science. Elsevier, pp. 575–616 (2006)
14. Gelman, A., Carlin, J.B., Stern, H.S., Rubin, D.B.: Bayesian Data Analysis, Second Edition (Texts in Statistical Science). Chapman & Hall/CRC, Boca Raton (2003)
15. George, A., Powell, W.B., Kulkarni, S.R.: Value function approximation using multiple aggregation for multiattribute resource management. J. Mach. Learn. Res. **9**, 2079–2111 (2008)
16. Gibbs, M.: Bayesian Gaussian Processes for Regression and Classification, dissertation. University of Cambridge, (1997)

17. Ginebra, J., Clayton, M.K.: Response surface bandits. J. R. Stat. Soc. Ser. B (Methodological) **57**, 771–784 (1995)
18. Gittins J., Jones D. (1974) A dynamic allocation index for the sequential design of experiments. In: Gani, J., Sarkadi, K., Vincze, I. (eds) Progress in Statistics. North-Holland, Amsterdam, pp. 241–266.
19. Gittins, J.C.: Bandit processes and dynamic allocation indices. J. R. Stat. Soc. Ser. B (Methodological) **41**, 148–177 (1979)
20. Gupta, S.S., Miescke, K.J.: Bayesian look ahead one-stage sampling allocations for selection of the best population. J. Stat. Plan. Inference, 54, 229–244. 40 Years of Statistical Selection Theory, Part I. (1996)
21. Hardle, W.K.: Applied Nonparametric Regression. Cambridge University Press, Cambridge (1992)
22. Hardle, W.K., Muller, M., Sperlich, S., Werwatz, A.: Nonparametric and Semiparametric Models. Springer, Berlin (2004)
23. Huang, D., Allen, T.T., Notz, W.I., Zeng, N.: Global optimization of stochastic black-box systems via sequential kriging meta-models. J. Glob. Optim. **34**, 441–466 (2006)
24. Juditsky, A., Nemirovski, A.: Functional aggregation for nonparametric regression. Ann. Stat. **28**, 681–712 (2000)
25. Kaelbling, L.P.: Learning in Embedded Systems. MIT Press, Cambridge (1993)
26. Kleinberg, R.: Nearly tight bounds for the continuum-armed bandit problem. In: Advances in Neural Information Processing Systems 17, MIT Press, pp. 697–704 (2005)
27. Mes, M.R., Powell, W.B., Frazier, P.I.: Hierarchical knowledge gradient for sequential sampling hierarchical knowledge gradient for sequential sampling. J. Mach. Learn. Res. **12**, 2931–2974 (2011)
28. Negoescu, D.M., Frazier, P.I., Powell, W.B.: The knowledge-gradient algorithm for sequencing experiments in drug discovery. INFORMS J. Comput. **23**, 346–363 (2011)
29. Nelson, B.L., Swann, J., Goldsman, D., Song, W.: Simple procedures for selecting the best simulated system when the number of alternatives is large. Oper. Res. **49**, 950–963 (2001)
30. Olafsson, S.: Chapter 21 metaheuristics, in Simulation. In: Henderson, S.G., Nelson, B.L. (eds.) vol. 13 of Handbooks in Operations Research and Management Science., pp. 633–654. Elsevier, (2006)
31. Powell, W.B.: Approximate Dynamic Programming: Solving the Curses of Dimensionality Wiley Series in Probability and Statistics. Wiley, Hoboken (2007)
32. Powell, W.B., Ryzhov, I.: Optimal Learning. Wiley, Philadelphia (2012)
33. Robbins, H., Monro, S.: A stochastic approximation method. Ann. Math. Stat. **22**, 400–407 (1951)
34. Ryzhov, I., Powell, W., Frazier, P.: The knowledge gradient algorithm for a general class of online learning problems, (2011)
35. Spall, J.C.: Introduction to Stochastic Search and Optimization. Wiley, New York (2003)
36. Sutton, R.S., Barto, A.G.: Introduction to Reinforcement Learning. MIT Press, Cambridge (1998)
37. Villemonteix, J., Vazquez, E., Walter, E.: An informational approach to the global optimization of expensive-to-evaluate functions. J. Glob. Optim. **44**, 509–534 (2009)