THE CORRELATED KNOWLEDGE GRADIENT FOR SIMULATION OPTIMIZATION OF CONTINUOUS PARAMETERS USING GAUSSIAN PROCESS REGRESSION

WARREN SCOTT, PETER FRAZIER, WARREN POWELL

Abstract. We extend the concept of the correlated knowledge-gradient policy for ranking and selection of a finite set of alternatives to the case of continuous decision variables. We propose an approximate knowledge gradient for problems with continuous decision variables in the context of a Gaussian process regression model in a Bayesian setting, along with an algorithm to maximize the approximate knowledge gradient. In the problem class considered, we use the knowledge gradient for continuous parameters to sequentially choose where to sample an expensive noisy function in order to find the maximum quickly. We show that the knowledge gradient for continuous decisions is a generalization of the efficient global optimization algorithm proposed by Jones, Schonlau, and Welch.

Key words. Model Calibration, Bayesian Global Optimization, Gaussian Process Regression, Knowledge Gradient, Expected Improvement

AMS subject classifications. 62L05 62L10 62L20

1. Introduction. Our goal is to find the global maximum of a real valued continuous function that is expensive to compute and that can only be evaluated with uncertainty. We need an algorithm that can give satisfactory results with as few function evaluations as possible. For this reason, we are willing to spend extra time deciding where we would like to evaluate the function next. This problem arises in applications such as simulation optimization, the design of machinery, medical diagnostics, biosurveillance, and the design of business processes.

We extend the concept of the knowledge-gradient policy for correlated beliefs (KGCB) described in [9] and [10], originally developed to find the best of a finite set of alternatives, to problems where we are trying to maximize over a multidimensional set of continuous variables. The KGCB policy maximizes the marginal value of a single measurement and has produced very promising results in discrete ranking and selection problems without requiring the use of any tunable parameters. In [10] the KGCB policy is used in a simulation optimization application to tune a set of continuous parameters which must be discretized to perform the search. However, the KGCB policy becomes computationally too expensive when it is necessary to discretize over a large multidimensional vector. We extend the knowledge gradient to multidimensional continuous problems, and then show that the knowledge gradient concept is at least competitive with, or outperforms, specialized algorithms for specific problems.

Although the concept for the knowledge gradient is very general, we choose to model the function to be optimized using Gaussian process regression with a squared exponential covariance function and model the noise in the observations as additive Gaussian noise. The knowledge gradient for continuous parameters (KGCP) policy that we propose extends the well known efficient global optimization algorithm in [16] to the case of noisy observations. When choosing a sampling decision, the KGCP accounts for the fact that an additional observation will update the regression function at unsampled decisions as well as at the sampling decision; the updated best decision will not necessarily be the current best decision or sampling decision.

This paper makes the following contributions: (1) We propose an approximation to the knowledge gradient for multidimensional continuous decision variables which can be efficiently computed; (2) We describe a gradient ascent algorithm that can be

used to maximize the knowledge gradient for continuous parameters without resorting to discretization; (3) We prove that, under mild conditions, the knowledge gradient for continuous parameters policy applied to maximizing a continuous function with observation noise will cause the uncertainty in the regression model to disappear in the limit; (4) We examine the competitive performance with sequential kriging, a widely used algorithm which lacks our theoretical guarantees, on a series of test functions.

This paper is organized as follows. Section 2 reviews the literature for continuous global optimization problems. Section 3 describes the Bayesian model capturing our prior belief in the function being optimized. We review the knowledge gradient for discrete alternatives, which guides measurements by computing the marginal value of information. Section 4 describes how the knowledge gradient can be computed for continuous measurements. The knowledge gradient for continuous parameters is then compared to the expected improvement in [16]. Our approach requires approximating the knowledge gradient as a continuous function, and we derive a gradient ascent algorithm for this purpose. In Section 5 we give mild conditions under which the posterior variance at each decision in the regression model will go to zero almost surely when using the knowledge gradient for continuous parameters policy for finding the global maximum of a function with observation noise. Finally, Section 6 compares the knowledge gradient for continuous parameters to sequential kriging optimization [15], which is a popular algorithm for determining sequential measurements in the presence of noise, on a set of test functions.

2. Literature Review. We briefly present and summarize some of the current approaches to maximizing an expensive function with observation noise. The applications are vast, and multiple research disciplines have addressed the problem. Simulation optimization covers gradient-based methods (see [32], [17], [39], [12], and [33]), direct search methods (see [39]), and metaheurstics (see [11]). The term model-based optimization can be used to categorize the fields of trust regions (see [29], [4], [3], and [6]), response surface methodology (see [2], [14], [26], [27], [28], and [31]), and the surrogate management framework (see [1]). Finally, Bayesian global optimization consists of algorithms which combine Bayesian models of the function with single-step look ahead criteria.

Bayesian global optimization takes a statistical approach to optimizing functions efficiently (see [35]). One of the first approaches in the field is [20] that approximates a one-dimensional function with a Wiener process and uses a probability of improvement criterion to choose the next point to sample. [40] uses the probability of improvement concept for higher dimensions in the P-algorithm. [45] as well as [25] and [21] also use a one-dimensional Wiener process but then use expected improvement criteria to choose the next point to sample; they discuss convergence in the case of no observation noise. For the case of no observation noise, [34] introduces the popular DACE (design and analysis of computer experiments) kriging model to approximate the expensive function; a kriging model is a method of interpolation based on random spatial processes (see [23], [5], [18], and [19]) and is referred to as Gaussian process regression in computer science (see [30]). [16] presents the efficient global optimization (EGO) algorithm for optimizing expensive functions without noise which combines a kriging model with an expected improvement criterion (also see [36] and [37]). Work has been done in [41] to prove convergence for an expected improvement algorithm in the case of no observation noise if the true function comes from a reproducing kernel Hilbert space generated by the covariance function. Another example of Bayesian global optimization is [13] which combines radial basis interpolation and a utility

function based on the uncertainty of the response surface weighted by how close the response surface's value at that point is to a specified target value.

Recent extensions of Bayesian global optimization explicitly account for observation noise, although limited convergence theory has been developed for the following algorithms. In [15], sequential kriging optimization (SKO) combines a kriging model with an expected improvement criterion which accounts for noisy functions observations; the expected improvement criterion is weighted by a term that favors decisions with higher uncertainty. One challenge of SKO, like many other Bayesian global optimization algorithms, is maximizing the expected improvement criterion to find the next sampling decision; the Nelder-Mead simplex method is suggested. [43] and [42] present an informational approach to global optimization (IAGO) which combines a kriging model, Monte Carlo, and other approximation techniques to estimate the distribution of the global minimizer of the function after an additional observation. The sampling decision is made by minimizing the entropy (which can be interpreted as uncertainty) of the global minimizer. The approaches in [7] and [8] address the issue of different levels of noise using an expected improvement criterion with kriging models found in [5] which allow for noisy observations.

3. The Model. We consider the following optimization problem

$$\text{argmax}_{\text{x}\in\mathcal{X}}\mu(\text{x}) \tag{3.1}$$

where $x \in \mathbb{R}^p$ is a decision vector, $\mathcal{X}$ is a compact feasible set of decisions, and $\mu : \mathbb{R}^p \to \mathbb{R}^1$ is a continuous function we wish to maximize. Let $\hat{y}^{n+1}$ be the sample observation of the sampling decision x^n for $n = 0, .., N-1$. The variance of an observation, given μ, at a decision x is $\lambda(x)$, and we assume $\lambda : \mathbb{R}^p \to \mathbb{R}^1$ is continuously differentiable over the domain $\mathcal{X}$ and is known. In practice, the variance of the observation noise is unknown but can be estimated. We assume $\hat{y}^{n+1}$ has a normal distribution centered around the true function,

$$\hat{y}^{n+1}|\mu, x^n \sim \mathcal{N}(\mu(x^n), \lambda(x^n)),$$

and $\hat{y}^1, ..., \hat{y}^{N+1}$ are independent given μ and $x^0, ..., x^N$. Our goal is to sequentially choose x^n at each iteration $n = 0, ..., N-1$ in order to approach the solution to (3.1) as quickly as possible.

Adopting a Bayesian framework, we start with some belief or information about the truth, μ. We treat μ as a random variable and assign it a Gaussian process (GP) prior density. μ^n is our updated mean of our random variable, given n observations. Then, for any $x^0, .., x^n \in \mathcal{X}$, our a priori distribution is $[\mu(x^0), ..., \mu(x^n)]^T \sim \mathcal{N}(\mu^0([x^0, ..., x^n]), \Sigma^0([x^0, ..., x^n])$ where $\mu^0([x^0, ..., x^n]) = \mathbb{E}([\mu(x^0), ..., \mu(x^n)]^T)$ and $\Sigma^0([x^0, ..., x^n]) = Cov([\mu(x^0), ..., \mu(x^n)]^T)$. Next we define a filtration $\mathcal{F}^n$ where $\mathcal{F}^n$ is the sigma-algebra generated by $x^0, \hat{y}^1, ..., x^{n-1}, \hat{y}^n$. We define $\mu^n([x^0, ..., x^n]) = \mathbb{E}([\mu(x^0), ..., \mu(x^n)]^T|\mathcal{F}^n)$ and $\Sigma^n([x^0, .., x^n]) = Cov([\mu(x^0), .., \mu(x^n)]^T|\mathcal{F}^n)$ for $x^0, ..., x^n \in \mathcal{X}$. In addition we use the notation $\Sigma^n(x^0, x^1) = Cov(\mu(x^0), \mu(x^1)|\mathcal{F}^n)$.

The multivariate normal distribution is a natural conjugate family when the observations come from a normal distribution with known variance. This means our posterior is also multivariate normal. Hence, conditioned on $\mathcal{F}^n$, $[\mu(x^0), ..., \mu(x^n)]^T \sim \mathcal{N}(\mu^n([x^0, ..., x^n]), \Sigma^n([x^0, ..., x^n]))$. Next we explain a method to assign the initial covariance between $\mu(x^0)$ and $\mu(x^1)$.

3.1. Covariance Structure. In order to specify the covariance matrix for our a priori distribution of μ at $x^0, ..., x^n \in \mathcal{X}$, it is sufficient to specify a covariance

function. Similar to [34] and [10], we assume a Gaussian covariance function. Letting x^0 and x^1 be arbitrary decisions in $\mathcal{X}$, we write,

$$Cov(\mu(x^0), \mu(x^1)) = \beta \exp(-\sum_{i=1}^{p} \alpha_i (x_i^0 - x_i^1)^2), \alpha > 0, \beta > 0, \tag{3.2}$$

where $\alpha \in \mathbb{R}^p$ is called the activity of μ and $\beta \in \mathbb{R}^1$ controls the uncertainty of our belief about μ. The initial covariance function given in (3.2) is a metric, meaning the covariance of two decisions decreases as the distance between them increases. The parameter α_i for $i = 1, ..., p$ is called the activity in dimension i and represents how smooth μ is in dimension i (see [16]). For example, a very small α_i would make the covariances bigger, indicating that μ is believed to be very smooth in dimension i. The key idea is that the true function should be positively correlated at nearby points. For example, if $\mu(x)$ is greater than $\mu^0(x)$, then, for small $\delta \in \mathbb{R}^p$, we should expect $\mu(x + \delta)$ to be greater than $\mu^0(x + \delta)$ as well, assuming μ is smooth. [30] explains that Gaussian processes with this covariance function are very smooth because they have mean square derivatives of all orders.

3.2. Updating Equations. After the first n sampling decisions, the distribution of $[\mu(x^0), ..., \mu(x^{n-1})]^T$ conditioned on $\mathcal{F}^n$ is multivariate normal and hence completely characterized by $\mu^n([x^0, ..., x^{n-1}])$ and $\Sigma^n([x^0, .., x^{n-1}])$, which can be calculated as follows in (3.6) and (3.7). For a fixed n, define the matrix $\Sigma^0 = \Sigma^0([x^0, ..., x^{n-1}])$ which can be calculated using (3.2). Given the assumptions in our model, we can use the Kalman filter equations in [24] or equivalently the Gaussian process regression equations given in [30] to compute the posterior distribution of μ given $\mathcal{F}^n$. We calculate the measurement residual $\tilde{y}^n$ and the residual covariance S^n as

$$\tilde{y}^n = \begin{bmatrix} \hat{y}^1 \\ \vdots \\ \hat{y}^n \end{bmatrix} - \begin{bmatrix} \mu^0(x^0) \\ \vdots \\ \mu^0(x^{n-1}) \end{bmatrix}, \tag{3.3}$$

$$S^n = \Sigma^0 + Diagonal([\lambda(x^0), ..., \lambda(x^{n-1})]). \tag{3.4}$$

We can then calculate the optimal Kalman gain using

$$K^n = \Sigma^0 [S^n]^{-1}. \tag{3.5}$$

Note that if the minimum value of the observation noises, λ^{min}, is strictly positive, $[S^n]^{-1}$ is well defined because the minimum eigenvalue of S^n is greater than λ^{min}. Let I_n be an $n \times n$ identity matrix. Finally, the updated expected values of μ at the first n sampled points, and the covariance matrix between μ at the first n sampled points, conditioned on $\mathcal{F}^n$, are given respectively by

$$\begin{bmatrix} \mu^n(x^0) \\ \vdots \\ \mu^n(x^{n-1}) \end{bmatrix} = \begin{bmatrix} \mu^0(x^0) \\ \vdots \\ \mu^0(x^{n-1}) \end{bmatrix} + K^n \tilde{y}^n, \tag{3.6}$$

$$\Sigma^n = (I_n - K^n)\Sigma^0. \tag{3.7}$$

The above equations update the distribution of μ at the first n sampling decisions conditioned on $\mathcal{F}^n$, but we also need to update the distribution of $\mu(x)$ conditioned

on $\mathcal{F}^n$, where $x \in X$ is an arbitrary decision variable that has not been sampled yet. We can do this with the following equations. Define $\bar{\Sigma}^0 = \Sigma^0([x^0, ..., x^{n-1}, x])$ and $\bar{\Sigma}^n = \Sigma^n([x^0, ..., x^{n-1}, x])$, and let $\vec{0}$ be a column vector of zeros. Our new optimal Kalman gain is given by

$$\bar{K}^n = \bar{\Sigma}^0 \begin{bmatrix} I_n \\ - \\ \vec{0}^T \end{bmatrix} [S^n]^{-1}. \tag{3.8}$$

We can now update μ^0 and $\bar{\Sigma}^0$ with the following equations,

$$\begin{bmatrix} \mu^n(x^0) \\ \vdots \\ \mu^n(x^{n-1}) \\ \mu^n(x) \end{bmatrix} = \begin{bmatrix} \mu^0(x^0) \\ \vdots \\ \mu^0(x^{n-1}) \\ \mu^0(x) \end{bmatrix} + \bar{K}^n \tilde{y}^n, \tag{3.9}$$

$$\bar{\Sigma}^n = (I_{n+1} - \bar{K}^n \begin{bmatrix} I_n & | & \vec{0} \end{bmatrix}) \bar{\Sigma}^0. \tag{3.10}$$

If we explicitly want the distribution of $\mu(x)$ conditioned on $\mathcal{F}^n$ at some arbitrary decision x we can pull out the pertinent formulae from (3.9) and (3.10);

$$\mu^n(x) = \mu^0(x) + \begin{bmatrix} \Sigma^0(x^0, x) & , \cdots , & \Sigma^0(x^{n-1}, x) \end{bmatrix} [S^n]^{-1} \tilde{y}^n, \tag{3.11}$$

$$\Sigma^n(x, x) = \Sigma^0(x, x) - \begin{bmatrix} \Sigma^0(x^0, x) & , \cdots , & \Sigma^0(x^{n-1}, x) \end{bmatrix} [S^n]^{-1} \begin{bmatrix} \Sigma^0(x^0, x) \\ \vdots \\ \Sigma^0(x^{n-1}, x) \end{bmatrix}. \tag{3.12}$$

Equation (3.11) is a linear smoother if $\mu^0(x) = 0 \ \forall x$ and is referred to as Gaussian process regression (GPR) in [30] and regressing kriging in [8]. There are also recursive equations equivalent to (3.9) and (3.10) which update μ^n and Σ^n (see [10]). [10] shows that after we have selected our sampling decision, x^n, but before we observe $\hat{y}^{n+1}$, our updated regression function is normally distributed conditioned on the information available at iteration n:

$$\begin{bmatrix} \mu^{n+1}(x^0) \\ \vdots \\ \mu^{n+1}(x^{n-1}) \\ \mu^{n+1}(x^n) \end{bmatrix} = \begin{bmatrix} \mu^n(x^0) \\ \vdots \\ \mu^n(x^{n-1}) \\ \mu^n(x^n) \end{bmatrix} + \tilde{\sigma}(\bar{\Sigma}^n, x^n) Z^{n+1}, \tag{3.13}$$

where $Z^{n+1} = \left(\hat{y}^{n+1} - \mu^n(x^n)\right) / \sqrt{\lambda(x^n) + \Sigma^n(x^n, x^n)}$, with

$$\tilde{\sigma}(\Sigma, x) \triangleq \frac{\Sigma e_x}{\sqrt{\lambda(x) + e_x^T \Sigma e_x}}; \tag{3.14}$$

here e_x is a column vector of zeros with a 1 at the row corresponding to decision x. It can be shown that $Z^{n+1} \sim \mathcal{N}(0, 1)$ because $\text{Var}(\hat{y}^{n+1} - \mu^n(x^n) | \mathcal{F}^n) = \lambda(x^n) + \Sigma^n(x^n, x^n)$.

3.3. The Knowledge-Gradient Policy. The knowledge-gradient policy as described in [10] for discrete $\mathcal{X}$ is the policy which chooses the next sampling decision by maximizing the expected incremental value of a measurement. The knowledge

gradient at x, which gives the expected incremental value of the information gained from a measurement at x, is defined as the following scalar field:

$$\nu^{KG,n}(x) \triangleq \mathbb{E}\left[\max_{u\in\mathcal{X}} \mu^{n+1}(u)\middle|\mathcal{F}^n, x^n = x\right] - \max_{u\in\mathcal{X}} \mu^n(u). \tag{3.15}$$

The knowledge-gradient policy chooses the sampling decision at time n by maximizing the knowledge gradient,

$$x^n \in \text{argmax}_{x\in\mathcal{X}} \nu^{KG,n}(x). \tag{3.16}$$

By construction, the knowledge gradient policy is optimal for maximizing the maximum of the predictor of the GP if only one decision is remaining. [10] shows that in the case of a finite set of decisions, the knowledge gradient policy samples every decision infinitely often as the number of sampling decisions goes to infinity; in other words, the knowledge gradient policy finds the best decision in the limit. In addition, [10] shows that the knowledge gradient policy is consistently competitive with or outperforms sequential kriging optimization (SKO) on several test functions.

The knowledge gradient can be explicitly computed when the feasible set of decisions, $\mathcal{X}$, is finite (see [10]). In the case where $\mathcal{X}$ is continuous, if p is small and $\mathcal{X}$ is bounded, then $\mathcal{X}$ can be discretized, allowing for the use of the technique in [10] for discrete decisions. However, the complexity of the calculation of this approximation of the knowledge gradient grows exponentially with the number of feasible decisions, $|x|$, because we must use a dense $|x| \times |x|$ covariance matrix in our calculation.

4. The Knowledge Gradient for Continuous Parameters. In this section we propose an approximation of the knowledge gradient that can be calculated and optimized when our feasible set of decisions is continuous. The approximation we propose can be calculated at a particular decision, x, along with its gradient at x, allowing us to use classical gradient-based search algorithms for maximizing the approximation. This strategy avoids the need to discretize the measurement space $\mathcal{X}$ into a large number of points to be evaluated. Furthermore, it scales to multidimensional parameter spaces which would be impossible to discretize.

We form the knowledge gradient for continuous parameters (KGCP) by replacing the maximum over $\mathcal{X} \subset \mathbb{R}^p$ with the maximum over $x^0, ..., x^n$, the first n sampling decisions and the current sampling decision,

$$\bar{\nu}^{KG,n}(x) \triangleq \mathbb{E}\left[\max_{i=0,..,n} \mu^{n+1}(x^i)\middle|\mathcal{F}^n, x^n = x\right] - \max_{i=0,..,n} \mu^n(x^i)|_{x^n=x}. \tag{4.1}$$

We define the knowledge gradient for continuous parameters policy, π^{KGCP}, as the policy which selects the next sampling decision by maximizing the knowledge gradient for continuous parameters,

$$x^n \in \text{argmax}_{x\in\mathcal{X}} \bar{\nu}^{KG,n}(x). \tag{4.2}$$

This approximation should improve as n increases and the maximization is taken over more terms. The first remark is that the knowledge gradient for continuous parameters is nonnegative. The proof follows from Jensen's inequality,

$$\begin{aligned}
\mathbb{E}\left[\max_{i=0,..,n} \mu^{n+1}(x^i)\middle|\mathcal{F}^n, x^n = x\right] &= \mathbb{E}\left[\max_{i=0,..,n} \mu^n(x^i) + \tilde{\sigma}_i(\bar{\Sigma}^n, x^n)Z^{n+1}\middle|\mathcal{F}^n, x^n = x\right] &(4.3)\\
&\geq \max_{i=0,..,n} \mu^n(x^i)|_{x^n=x} + \tilde{\sigma}_i(\bar{\Sigma}^n, x^n)\mathbb{E}\left[Z^{n+1}\middle|\mathcal{F}^n, x^n = x\right] &(4.4)\\
&= \max_{i=0,..,n} \mu^n(x^i)|_{x^n=x}.
\end{aligned}$$

In (4.3) we substituted in the recursive update for $\mu^{n+1}(x^i)$ given in (3.13). $\tilde{\sigma}_i(\Sigma, x)$ is the i^{th} element of $\tilde{\sigma}(\Sigma, x)$ which is defined in (3.14). In (4.4) we use Jensen's inequality with the convex function $\phi(z) = \max_{i=0,..,n} \mu^n(x^i) + \tilde{\sigma}_i(\bar{\Sigma}^n, x^n)z$ where $\mu^n(x^i)$ and $\tilde{\sigma}_i(\bar{\Sigma}^n, x^n)$ are constants since they are measurable with respect to $\mathcal{F}^n$.

Also, comparing the terms that depend on x in the knowledge gradient and the knowledge gradient for continuous parameters, we easily see that

$$\mathbb{E}\left[\max_{i=0,..,n} \mu^{n+1}(x^i)|\mathcal{F}^n, x^n = x\right] \leq \mathbb{E}\left[\max_{u\in\mathcal{X}} \mu^{n+1}(u)|\mathcal{F}^n, x^n = x\right]. \tag{4.5}$$

This fact follows trivially because the maximization in the left term is over a subset of the set maximized over in the right term. Initially, at time $n = 0$, the knowledge gradient for continuous parameters becomes

$$\bar{\nu}^{KG,0}(x) = \mathbb{E}[\mu^1(x^0)|\mathcal{F}^0, x^0 = x] - \mu^0(x^0)|_{x^0=x} = \mu^0(x) - \bar{\mu}^0(x) = 0.$$

This shows the KGCP policy is indifferent about the first sampling decision. At time $n = 1$, (4.2) becomes

$$x^1 \in \text{argmax}_{x\in\mathcal{X}} \left(\mathbb{E}[\max_{i=0,1} \mu^2(x^i)|\mathcal{F}^1, x^1 = x] - \max_{i=0,1} \mu^1(x^i)|_{x^1=x}\right).$$

At this point there is a trade-off between exploring and exploiting in our objective. Implicitly, the algorithm would like to exploit, or sample near a current maximum of μ^n; this seems likely to increase the maximum of μ^n. However, the algorithm would also like to explore, i.e. sample far away from any of the previous decisions; these decisions have more uncertainty and are less correlated with the current maximum of μ^n.

4.1. Comparison to the Expected Improvement of EGO. Efficient Global Optimization (EGO) is a method developed in [16] to optimize functions when there is no observation noise. For function maximization, EGO uses the expected improvement criterion, $\mathbb{E}[I^n(x)|\mathcal{F}^n]$, where the improvement given the information available at time n is defined to be the following random variable:

$$I^n(x) = \max\left(\mu^{n+1}(x) - \max_{i=1,..,n} \hat{y}^i, 0\right).$$

In [16], the EGO expected improvement is only defined in the case of no observation noise, where $\lambda(\cdot) = 0$. In this case, the knowledge gradient for continuous parameters is less than or equal to the EGO expected improvement criterion. In fact, if the second maximization term in the knowledge gradient for continuous parameters in (4.1) were over $i = 0, .., n-1$, the knowledge gradient for continuous parameters would be equivalent to the expected improvement in the case of no observation noise.

PROPOSITION 4.1. *In the case of no observation noise,* $\bar{\nu}^{KG,n}(x) \leq \mathbb{E}[I^n(x)|\mathcal{F}^n]$. *Furthermore,* $\mathbb{E}[I^n(x)|\mathcal{F}^n] = \mathbb{E}\left[\max_{i=0,..,n} \mu^{n+1}(x^i)|\mathcal{F}^n, x^n = x\right] - \max_{i=0,..,n-1} \mu^n(x^i)$.

Proof:

$$\begin{aligned}
\bar{\nu}^{KG,n}(x) &= \mathbb{E}\left[\max_{i=0,..,n} \mu^{n+1}(x^i)|\mathcal{F}^n, x^n = x\right] - \max_{i=0,..,n} \mu^n(x^i)|_{x^n=x} \\
&\leq \mathbb{E}\left[\max_{i=0,..,n} \mu^{n+1}(x^i)\middle|\mathcal{F}^n, x^n = x\right] - \max_{i=0,..,n-1} \mu^n(x^i) \\
&= \mathbb{E}\left[\max\left(\mu^{n+1}(x^n), \max_{i=0,..,n-1} \mu^n(x^i)\right)\middle|\mathcal{F}^n, x^n = x\right] - \max_{i=0,..,n-1} \mu^n(x^i) \quad (4.6) \\
&= \mathbb{E}\left[\max\left(\mu^{n+1}(x^n), \max_{i=1,..,n} \hat{y}^i\right)\middle|\mathcal{F}^n, x^n = x\right] - \max_{i=1,..,n} \hat{y}^i \\
&= \mathbb{E}\left[\max\left(\mu^{n+1}(x^n) - \max_{i=1,..,n} \hat{y}^i, 0\right)\middle|\mathcal{F}^n, x^n = x\right] \\
&= \mathbb{E}[I^n(x)|\mathcal{F}^n]. \quad (4.7)
\end{aligned}$$

In (4.6) we used the fact that, conditioned on $\mathcal{F}^n$, $\hat{y}^{i+1} = \mu^n(x^i) = \mu^{n+1}(x^i)$ for $i = 0, ..., n-1$ since there is no observation noise. □

The EGO algorithm maximizes the expected improvement given in (4.7) at each iteration which is similar to maximizing the knowledge gradient for continuous parameters at each iteration when there is no observation noise.

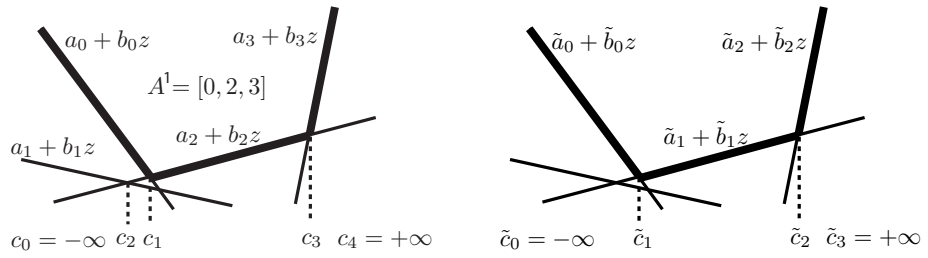
4.2. Calculation of the Knowledge Gradient for Continuous Parameters. We will first show how to calculate the knowledge gradient for continuous parameters, and then derive the gradient of this continuous function that can be used in a steepest ascent algorithm. The knowledge gradient for continuous parameters in (4.1) can be efficiently calculated at a particular $x \in \mathcal{X}$ by using the two algorithms in [10], which we will now summarize. We define the pairs (a_i, b_i) for $i = 0, ..., n$ as the sorted pairs $(\mu^n(x^i), \tilde{\sigma}_i(\bar{\Sigma}^n, x^n))$ conditioned on $\mathcal{F}^n$ and $x^n = x$ for $i = 0, ..., n$. The pairs (a_i, b_i) are sorted such that $b_i \leq b_{i+1}$ for $i = 0, ..., n-1$. If there exists some $i \neq j$ such that $b_i = b_j$ and $a_i \leq a_j$, then the pair (a_j, b_j) dominates (a_i, b_i) and you add (a_i, b_i) to a list of *initially* dominated lines. The a_i's are the intercepts and the b_i's are the slopes of the lines in Figure 4.1(a). Furthermore we define A^0 as the index map such that $(a_i, b_i) = (\mu^n(x^{A_i^0}), \tilde{\sigma}_{A_i^0}(\bar{\Sigma}^n, x^n))$. For a fixed $x^n = x$, a_i and b_i are $\mathcal{F}^n$ measurable and hence constants. We now simplify the first term in the knowledge gradient for continuous parameters,

$$\begin{aligned}
\mathbb{E}\left[\max_{i=0,...,n} \mu^{n+1}(x^i)\middle|\mathcal{F}^n, x^n = x\right] &= \mathbb{E}\left[\max_{i=0,...,n} \mu^n(x^i) + \tilde{\sigma}_i(\bar{\Sigma}^n, x^n)Z^{n+1}\middle|\mathcal{F}^n, x^n = x\right] \quad (4.8) \\
&= \mathbb{E}\left[\max_{i=0,...,n} a_i + b_i Z\right]. \quad (4.9)
\end{aligned}$$

In (4.8) we substituted in the recursive update for $\mu^n(x^i)$ given in (3.13). We next summarize the two algorithms in [10] which show how to efficiently calculate the term in (4.9).

Algorithm 1 is a scan-line algorithm that replaces the maximization in (4.9) with a piecewise linear function using indicator functions. In Algorithm 1, A^1 is called the accept set and is a vector of indices which keeps track of all the i's such that line $a_i + b_i z$ is part of the epigraph shown in Figure 4.1(a). We keep track of the values of z where the lines intersect in a vector c. c_{i+1} is the largest value of z such that line $a_i + b_i z$ is part of the epigraph shown in Figure 4.1(a). In terms of the lines in the accept set A^1, $c_{1+A_i^1}$ is the intersection of $a_{A_i^1} + b_{A_i^1} z$ and $a_{A_{i+1}^1} + b_{A_{i+1}^1} z$. Solving for the z such that these lines intersect we get $c_{1+A_i^1} = (a_{A_i^1} - a_{A_{i+1}^1})/(b_{A_{i+1}^1} - b_{A_i^1})$ for $i = 1, ..., \tilde{n}$, where $\tilde{n}$ is the length of A^1 minus one. Also we set $c_0 = -\infty$ and

$c_{n+1} = +\infty$. For convenience, we define $\tilde{a}_i = a_{(A_i^1)}$, $\tilde{b}_i = b_{(A_i^1)}$, $\tilde{c}_{i+1} = c_{(1+A_i^1)}$, and $\tilde{c}_0 = -\infty$ for $i = 0, ..., \tilde{n}$. Algorithm 1 efficiently calculates constants $\tilde{c}_0, ..., \tilde{c}_{\tilde{n}+1}$ and the vector of indices, A^1, so that a function of the form $f(z) = \max_{i=0,...,n} a_i + b_i z$ can be rewritten as $f(z) = \sum_{i=0}^{\tilde{n}} (a_{A_i^1} + b_{A_i^1} z) \mathbf{1}_{[\tilde{c}_i, \tilde{c}_{i+1})}(z)$. The algorithm is outlined in Figure 4.2, using the convention that the first index of a vector is zero.



(a) A visualization of Algorithm 1. (b) The output of Algorithm 1 with new indices.

FIG. 4.1. *Algorithm 1 is a scan line algorithm to re-express* $f(z) = \max_{i=0,...,n} a_i + b_i z$ *as* $f(z) = \sum_{i=0}^{\tilde{n}} (\tilde{a}_i + \tilde{b}_i z) \mathbf{1}_{[\tilde{c}_i, \tilde{c}_{i+1})}(z)$.

Next, Algorithm 2 from [10] shows how to simplify the expectation in (4.10) to (4.11), which is something we can easily compute.

$$\mathbb{E}\left[\max_{i=0,...,n} a_i + b_i Z\right] = \mathbb{E}\left[\sum_{i=0}^{\tilde{n}} \left(a_{A_i^1} + b_{A_i^1} Z\right) \mathbf{1}_{[\tilde{c}_i, \tilde{c}_{i+1})}(Z)\right] \tag{4.10}$$

$$= \sum_{i=0}^{\tilde{n}} \left[a_{A_i^1} \mathbb{P}[Z \in [\tilde{c}_i, \tilde{c}_{i+1})] + b_{A_i^1} \mathbb{E}[Z \mathbf{1}_{[\tilde{c}_i, \tilde{c}_{i+1})}(Z)]\right]$$

$$= \sum_{i=0}^{\tilde{n}} \left[a_{A_i^1} \left(\Phi(\tilde{c}_{i+1}) - \Phi(\tilde{c}_i)\right) + b_{A_i^1} \left(\phi(\tilde{c}_i) - \phi(\tilde{c}_{i+1})\right)\right] \tag{4.11}$$

In (4.11), $\phi(\cdot)$ and $\Phi(\cdot)$ are the pdf and cdf of a standard normal random variable, respectively.

4.3. The Gradient of the Knowledge Gradient for Continuous Parameters. Next, we show how to calculate the gradient of the knowledge gradient for continuous parameters, $\nabla_x \bar{\nu}^{KG,n}(x)$, at a fixed $x \in \mathcal{X}$. This will allow us to use gradient ascent to maximize the knowledge gradient for continuous parameters. Let $A = A^0[A^1]$, meaning $A_i = A^0_{A_i^1}$; A is now a reordered index set. For example, if $A^0 = [2, 1, 0]$ and $A^1 = [0, 2, 1]$, then $A = [2, 0, 1]$. A contains the indices i such that $(\mu^n(x^{A_i^0}) + \tilde{\sigma}_{A_i^0}(\bar{\Sigma}^n, x^n))z$ is part of the epigraph of Figure 4.1(b) for some value of z.

PROPOSITION 4.2. *The gradient of the first term in* (4.1) *is given by*

$$\nabla_x \mathbb{E}\left[\max_{i=0,..,n} \mu^{n+1}(x^i) \middle| \mathcal{F}^n, x^n = x\right]$$

$$= \sum_{i=0}^{\tilde{n}} \left[\left(\nabla_{x^n} \mu^n(x^{A_i})\right) \left(\Phi(\tilde{c}_{i+1}) - \Phi(\tilde{c}_i)\right) + \left(\nabla_{x^n} \tilde{\sigma}_{A_i}(\bar{\Sigma}^n, x^n)\right) \left(\phi(\tilde{c}_i) - \phi(\tilde{c}_{i+1})\right)\right]$$

$$+ \sum_{i=0}^{\tilde{n}} \left[\left(\mu^n(x^{A_i}) + \tilde{\sigma}_{A_i}(\bar{\Sigma}^n, x^n)\tilde{c}_{i+1}\right) \phi(\tilde{c}_{i+1}) \nabla_{x^n} \tilde{c}_{i+1} - \left(\mu^n(x^{A_i}) + \tilde{\sigma}_{A_i}(\bar{\Sigma}^n, x^n)\tilde{c}_i\right) \phi(\tilde{c}_i) \nabla_{x^n} \tilde{c}_i\right].$$

```
(01) c_0 = −∞, c_{n+1} = +∞, A^1 = [0]
(02) for i = 1 : n
(03)     if (a_i, b_i) not initially dominated
(04)         loopdone = false
(05)         while loopdone == false
(06)             j = A^1(end)
(07)             c_{j+1} = (a_j − a_i)/(b_i − b_j)
(08)             if length(A^1) ≠ 1 & c_{j+1} ≤ c_{k+1} where k = A^1(end − 1)
(09)                 Delete last element in A^1.
(10)             else add i to the end of A^1.
(11)                 loopdone = true
(12)             end
(13)         end
(14)     end
(15) end
```

FIG. 4.2. *Summary of Algorithm 1 from [10].*

Proof:

$$\nabla_{x^n}\mathbb{E}\left[\max_{i=0,..,n}\mu^{n+1}(x^i)\middle|\mathcal{F}^n, x^n = x\right]$$

$$= \nabla_{x^n}\sum_{i=0}^{\tilde{n}}\left[\mu^n(x^{A_i})\left(\Phi(\tilde{c}_{i+1}) - \Phi(\tilde{c}_i)\right) + \tilde{\sigma}_{A_i}(\bar{\Sigma}^n, x^n)\left(\phi(\tilde{c}_i) - \phi(\tilde{c}_{i+1})\right)\right] \tag{4.12}$$

$$= \sum_{i=0}^{\tilde{n}}\left[\left(\nabla_{x^n}\mu^n(x^{A_i})\right)\left(\Phi(\tilde{c}_{i+1}) - \Phi(\tilde{c}_i)\right) + \left(\nabla_{x^n}\tilde{\sigma}_{A_i}(\bar{\Sigma}^n, x^n)\right)\left(\phi(\tilde{c}_i) - \phi(\tilde{c}_{i+1})\right)\right]$$

$$+ \sum_{i=0}^{\tilde{n}}\left[\mu^n(x^{A_i})\nabla_{x^n}\left(\Phi(\tilde{c}_{i+1}) - \Phi(\tilde{c}_i)\right) + \tilde{\sigma}_{A_i}(\bar{\Sigma}^n, x^n)\nabla_{x^n}\left(\phi(\tilde{c}_i) - \phi(\tilde{c}_{i+1})\right)\right] \tag{4.13}$$

$$= \sum_{i=0}^{\tilde{n}}\left[\left(\nabla_{x^n}\mu^n(x^{A_i})\right)\left(\Phi(\tilde{c}_{i+1}) - \Phi(\tilde{c}_i)\right) + \left(\nabla_{x^n}\tilde{\sigma}_{A_i}(\bar{\Sigma}^n, x^n)\right)\left(\phi(\tilde{c}_i) - \phi(\tilde{c}_{i+1})\right)\right]$$

$$+ \sum_{i=0}^{\tilde{n}}\left[\left(\mu^n(x^{A_i}) + \tilde{\sigma}_{A_i}(\bar{\Sigma}^n, x^n)\tilde{c}_{i+1}\right)\phi(\tilde{c}_{i+1})\nabla_{x^n}\tilde{c}_{i+1} - \left(\mu^n(x^{A_i}) + \tilde{\sigma}_{A_i}(\bar{\Sigma}^n, x^n)\tilde{c}_i\right)\phi(\tilde{c}_i)\nabla_{x^n}\tilde{c}_i\right].$$

Equation (4.12) is just the gradient of (4.11). In (4.13) we used the product rule because $c_0, ..., c_{n+1}$ all depend on x^n. In the last line we use the fact that $\frac{\partial}{\partial x}\Phi(f(x)) = \phi(f(x))\frac{\partial}{\partial x}f(x)$ and $\frac{\partial}{\partial x}\phi(f(x)) = -\phi(f(x))f(x)\frac{\partial}{\partial x}f(x)$ to differentiate the second term. The first term in the final equation is analogous to (4.11) with the scalars $\mu^n(x^i)$ and $\tilde{\sigma}_i(\bar{\Sigma}^n, x^n)$ replaced with the vectors $\nabla_{x^n}\mu^n(x^i)$ and $\nabla_{x^n}\tilde{\sigma}_i(\bar{\Sigma}^n, x^n)$. □

The calculation of $\nabla_{x^n}\tilde{c}_i$ for $i = 0, ..., \tilde{n}+1$ is relatively straightforward. An equivalent equation for the $\tilde{c}_i$'s which are output from Algorithm 1 is $\tilde{c}_i = \frac{\tilde{a}_{i-1} - \tilde{a}_i}{\tilde{b}_i - \tilde{b}_{i-1}}$ for $i = 1, ..., \tilde{n}$ with $\tilde{c}_0 = -\infty$ and $\tilde{c}_{\tilde{n}+1} = +\infty$. Then using the quotient rule we can calculate the following:

$$\nabla_{x^n}\tilde{c}_i = \begin{cases} \frac{(\tilde{b}_i - \tilde{b}_{i-1})(\nabla\tilde{a}_{i-1} - \nabla\tilde{a}_i) - (\tilde{a}_{i-1} - \tilde{a}_i)(\nabla\tilde{b}_i - \nabla\tilde{b}_{i-1})}{(\tilde{b}_i - \tilde{b}_{i-1})^2}, & \text{for } i = 1, ..., \tilde{n} \\ \vec{0}, & \text{for } i = 0, \tilde{n}+1. \end{cases} \tag{4.14}$$

As long as we can calculate $\nabla_{x^n}\mu^n(x^i)$ and $\nabla_{x^n}\tilde{\sigma}_i(\bar{\Sigma}^n, x^n)$ for $i = 0, ..., n$, we can calculate the expression in Proposition 4.2 and the gradient of the knowledge gradient

for continuous parameters. The equations for these values are expressed in the next two lemmas.

LEMMA 4.3.

$$\nabla_{x^n}\mu^n(x^i) = \begin{cases} \vec{0}, & \text{if } i < n \\ \nabla_{x^n}\mu^0(x^n) + J^n[S^n]^{-1}\tilde{y}^n, & \text{if } i = n, \end{cases}$$

where we let J^n be the following matrix of first-order partial derivatives,

$$J^n = \left[\nabla_{x^n}\Sigma^0(x^0, x^n) \quad , \cdots, \quad \nabla_{x^n}\Sigma^0(x^{n-1}, x^n)\right] \tag{4.15}$$

$$= 2\begin{bmatrix} \alpha_1(x_1^0 - x_1^n)\Sigma^0(x^0, x^n) & \cdots & \alpha_1(x_1^{n-1} - x_1^n)\Sigma^0(x^{n-1}, x^n) \\ \vdots & \ddots & \vdots \\ \alpha_p(x_p^0 - x_p^n)\Sigma^0(x^0, x^n) & \cdots & \alpha_p(x_p^{n-1} - x_p^n)\Sigma^0(x^{n-1}, x^n) \end{bmatrix}. \tag{4.16}$$

Proof: Given in Appendix 8.1.

LEMMA 4.4.

$$\nabla_{x^n}\tilde{\sigma}_i(\bar{\Sigma}^n, x^n) = \frac{B\nabla_{x^n}e_{x^i}^T\bar{\Sigma}^n e_{x^n} - e_{x^i}^T\bar{\Sigma}^n e_{x^n}\nabla_{x^n}B}{B^2},$$

where $B \triangleq \sqrt{\lambda(x^n) + e_{x^n}^T\bar{\Sigma}^n e_{x^n}}$ and

$$\nabla_{x^n}e_{x^i}^T\bar{\Sigma}^n e_{x^n} = \begin{cases} 2\text{DIAG}(\alpha)(x^i - x^n)\Sigma^0(x^i, x^n) - J^n[S^n]^{-1}\Sigma^0 e_{x^i}, & \text{if } i < n \\ -2J^n[S^n]^{-1}\begin{bmatrix} \Sigma^0(x^0, x^n) \\ \vdots \\ \Sigma^0(x^{n-1}, x^n) \end{bmatrix}, & \text{if } i = n \end{cases}$$

and

$$\nabla_{x^n}B = \frac{1}{2}(\lambda(x^n) + \Sigma^n(x^n, x^n))^{-\frac{1}{2}}\left(\nabla_{x^n}\lambda(x^n) - 2J^n[S^n]^{-1}\begin{bmatrix} \Sigma^0(x^0, x^n) \\ \vdots \\ \Sigma^0(x^{n-1}, x^n) \end{bmatrix}\right).$$

Proof: Given in Appendix 8.2.

4.4. Maximizing the Knowledge Gradient for Continuous Parameters. We begin by giving an illustrative example of the knowledge gradient for continuous parameters on a one-dimensional Gaussian process with normally distributed observation noise with a variance of 0.1. Figure 4.3(a) shows the results of the estimate of the function after four observations along with the actual observations. Figure 4.3(b) shows both the knowledge gradient for continuous parameters and the exact knowledge gradient over a finely discretized set of decisions. The knowledge gradient is larger at decisions with more uncertainty as well as points where the estimate of the function is larger. We can see that the knowledge gradient is nonconcave and seems to have local minima near previously sampled points. Furthermore, many of the local maxima appear to be approximately halfway between previously sampled points.

In Figure 4.3(c) and 4.3(d) we show the estimate of the function and knowledge gradient after nine observations. Again the knowledge gradient is not concave but many of the local maxima appear to be approximately halfway between previously

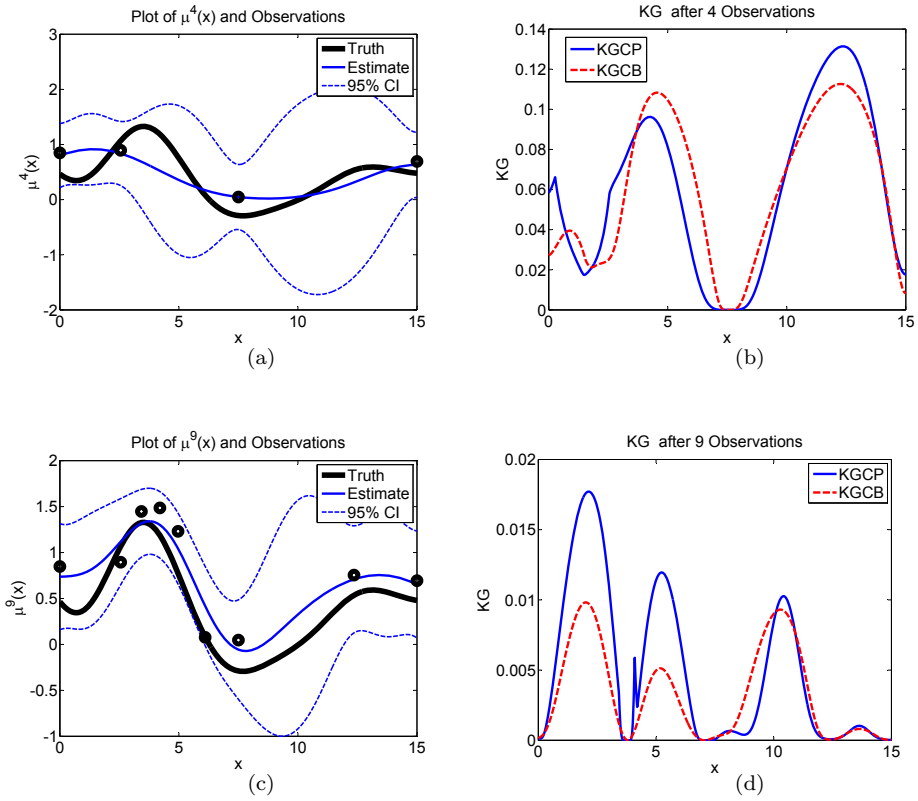


FIG. 4.3. *(a) The estimate of the function along with the 95% confidence intervals of the estimate after 4 observations. (b) The knowledge gradient for continuous parameters (KGCP) and exact knowledge gradient over a finely discretized set of decisions (KGCB) after 4 observations. (c) The estimate of the function after 9 observations. (d) The knowledge gradient after 9 observations.*

sampled points. In higher dimensions, a gradient ascent algorithm started multiple times is appropriate for approximately maximizing a nonconcave continuous function.

We now have an objective that can be quickly evaluated along with its gradient at any decision x. We propose using a multi-start gradient ascent algorithm with constraints for the domain. Heuristically, as suggested above, there is likely to be a local maximum roughly halfway between two previously sampled points. Furthermore, we have a good guess at a starting step size that will keep our algorithm looking in the region between these two previously sampled points based on the distance between the two points. We can calculate all the midpoints between the set of sampled points and use them as starting points of our gradient ascent with a fixed step size chosen such that the magnitude of the first step is one fourth of the Euclidian distance between the two corresponding previously sampled points. We also choose to start the gradient ascent algorithm at the previously sampled decisions. These points are likely to be very close to a local minimum and are thus reasonable starting locations for a gradient ascent algorithm, although a reasonable starting step size is more ambiguous. We can then take the maximum over all of the restarts to approximately get the overall maximum of the knowledge gradient for continuous parameters. We perform $\binom{n}{2} + n$

restarts which may become computationally expensive at n grows large. Alternatively we could maximize KGCP over a set of candidate points chosen by an LHS design or use a genetic algorithm (see [8]). It is worth noting that it is not critical to get the exact maximum of the knowledge gradient for continuous parameters in order to determine the next sampling decision. There are likely several distinct points that are worth sampling and it may be acceptable if on one iteration the algorithm chooses a point which does not exactly maximize the knowledge gradient for continuous parameters.

4.5. The KGCP Policy. We now give an outline of the KGCP policy.

KGCP policy
(1) **for** $n = 0, ..., N-1$
(2) Choose sampling decision: $x^n \in \arg\max_{x\in\mathcal{X}} \bar{\nu}^{\mathrm{KG},n}(x)$ using Section 4.4.
(3) Get noisy observation $\hat{y}^{n+1}$ of function at x^n.
(4) Update μ^{n+1} and Σ^{n+1} using (3.9) and (3.10).
(6) **end**
(7) Implement $x^\star \in \arg\max_{x\in\mathcal{X}} \mu^N(x)$.

In line 2 we choose the sampling decision by maximizing the knowledge gradient for continuous parameters defined in (4.1). This maximization should be approximated by using the algorithm in Section 4.4. Also, the maximization in line 7 to find the implementation decision cannot be explicitly solved either. We approximate the solution using a multistart gradient ascent algorithm with the same starting points used in Section 4.4. The gradient of $\mu^N(x)$ can be evaluated using Lemma 4.3. If no prior knowledge about the parameters is available, an initial phase of sampling decisions chosen following a Latin hypercube design can be run before starting the KGCP policy as suggested in a similar context in [16].

In general we will not be given the parameters of the covariance function, α and β, the variance of observation noise, $\lambda()$, or the mean of the initial prior distribution on μ, $\mu^0()$. If these parameters are not known, a step should be added before line 2 for estimating the covariance function parameters using MLE, maximum a posterior estimation (see [30]), or robust parameter estimation (see [38]). For example, we can approximately maximize the likelihood over the parameters by using patternsearch() in Matlab started at multiple points chosen by a Latin hypercube sampling (LHS) design using the command lhsdesign().

5. Convergence. In this section we show that, although the KGCP can be regarded as a near-sighted objective for finding the maximum of $\mu(x)$, the KGCP policy searches enough so that uncertainty of the regression function converges to zero almost surely for each decision as the number of sampling decisions and observations increases to infinity. Note that additional conditions would need to be specified before making the claim about the consistency of the posterior and finding the maximum of $\mu(x)$ almost surely in the limit. The proof is based on the fact that the knowledge gradient for continuous parameters of each decision converges to zero as the number of iterations of the algorithm goes to infinity. We then show that this implies that the conditional variance of μ at every observation converges to zero; in other words, we become certain of μ at every point. We define $Var^n[\cdot]$, $Cov^n[\cdot]$, and $Corr^n[\cdot]$ as $Variance[\cdot|\mathcal{F}^n]$, $Covariance[\cdot|\mathcal{F}^n]$, and $Correlation[\cdot|\mathcal{F}^n]$, respectively. For simplicity in this section we assume the variance of the observation noise is a constant. Our presentation will need the following assumptions:

ASSUMPTION 5.0.1. *$\lambda(x) = \lambda > 0$, $\mu^0(x) = \mu^0$, and the estimates of α, β, λ, and μ^0 are fixed.*

ASSUMPTION 5.0.2. $\limsup_{n\to\infty} |\mu^n(x) - \mu^n(u)|$ *is bounded for every* $x, u \in \mathcal{X}$ *almost surely.*

ASSUMPTION 5.0.3. *For any* $x \neq u$, $\exists c$ *s.t.* $\limsup_{n\to\infty} |Corr^n[\mu(x), \mu(u)]| \leq c < 1$ *almost surely.*

ASSUMPTION 5.0.4. *We can exactly maximize the KGCP;* $x^n \in argmax_{x\in\mathcal{X}} \bar{\nu}^{KG,n}(x)$.

PROPOSITION 5.1. *For every sample path, the knowledge gradient for continuous parameters of a decision* x, $\bar{\nu}^{KG,n}(x)$, *converges to zero if the conditional variance of* $\mu(x)$ *converges to zero.*

Proof: We first need an upper bound on the knowledge gradient for continuous parameters. We show in Appendix 8.3 that

$$\bar{\nu}^{\mathrm{KG},n}(x) \leq \sqrt{\frac{2\beta Var^n[\mu(x)]}{\pi\lambda}}. \tag{5.1}$$

Combining the fact that the knowledge gradient for continuous parameters is nonnegative and that the upper bound of the knowledge gradient for continuous parameters in (5.1) decreases to zero as $Var^n[\mu(x)] \to 0$, we obtain the desired result. □

The next proposition provides a way to put an upper bound on the conditional variance of μ near an accumulation point, x^{acc}, of the sampling decisions. Figure 5.1 has a diagram of the points being considered. x^{acc} is an accumulation point of the sampling decisions. x^d is an arbitrary fixed point in an open ball centered around x^{acc} with radius ϵ; we are interested in $Var[\mu(x^d)]$. x^{mult} is a point we consider measuring multiple times. x^{near} is a point which is closer to x^d than x^{mult} is close to x^d in terms of the initial covariance; formally, $\Sigma^0(x^{mult}, x^d) \leq \Sigma^0(x^{near}, x^d)$. We denote an open ball centered at a with radius ϵ as $B(a, \epsilon) = \{x | d(x, a) < \epsilon\}$.

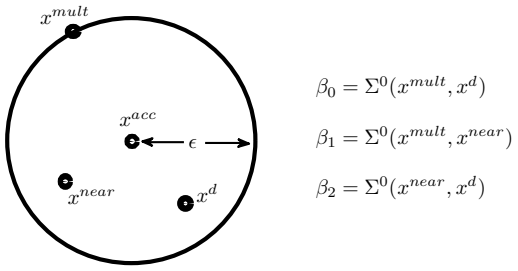


FIG. 5.1. *A diagram of the points:* x^{acc} *is an accumulation point;* x^{mult} *is a point being measured multiple times;* x^{near} *is a point near to* x^d *we are considering to measure;* x^d *is an arbitrary fixed point in the open ball centered at* x^{acc}.

PROPOSITION 5.2. *Fix* $\epsilon > 0$ *and consider an arbitrary point* $x^d \in B(x^{acc}, \epsilon)$, *where* $B(x^{acc}, \epsilon)$ *is an open ball centered at* x^{acc} *with radius* ϵ. *If we have measured* n *points in the ball* $B(x^{acc}, \epsilon)$, *an upper bound on the conditional variance of* $\mu(x^d)$ *can be constructed by hypothetically measuring one particular point* x^{mult} n *times, where* x^{mult} *satisfies* $\Sigma^0(x^{mult}, x^d) \leq \Sigma^0(x, x^d), \forall x \in B(x^{acc}, \epsilon)$. *Furthermore the upper bound on the conditional variance of* $\mu(x^d)$ *is* $\beta - (\Sigma^0(x^{mult}, x^d))^2 \frac{n}{\beta n + \lambda}$ *for every sample path.*

Proof: Sketch of proof (See Appendix 8.4 for full proof): We wish to find an upper bound on the conditional variance of $\mu(x^d)$ which will converge to zero as $n \to \infty$

and $\epsilon \to 0$. The ordering of the decision-observation pairs can be changed without altering the conditional variance of $\mu(x^d)$, and the conditional variance of $\mu(x^d)$ is a decreasing sequence. Therefore, after we have measured n points in $B(x^{acc}, \epsilon)$, $max_{x_0,...,x_{n-1} \in B(x^{acc},\epsilon)} Var^n[\mu(x^d)]$ is an upper bound on the conditional variance of $\mu(x^d)$; we have ignored the decisions outside of $B(x^{acc}, \epsilon)$ because they would only lower the conditional variance more. We define the policy π^{mult} which sets $x^0 = \cdots = x^{n-1} = x^{mult}$. We can derive that under the policy π^{mult}, $Var^n[\mu(x)] = \beta - (\Sigma^0(x^{mult}, x))^2 \frac{n}{\beta n + \lambda}$.

First consider the change $Var^n[\mu(x^d)] - Var^{n+1}[\mu(x^d)]$ under π^{mult} if we have measured x^{mult} n times and then measure x^{mult} one more time. We define $\beta_0 = \Sigma^0(x^{mult}, x^d)$. The decrease in the conditional variance of $\mu(x^d)$ from measuring x^{mult} once more is

$$Var^n[\mu(x^d)] - Var^{n+1}[\mu(x^d)] = \frac{\beta_0^2 \lambda}{((n+1)\beta + \lambda)(n\beta + \lambda)}. \tag{5.2}$$

Second we consider measuring the change in $Var^n[\mu(x^d)] - Var^{n+1}[\mu(x^d)]$ if we have measured x^{mult} n times and then measure x^{near} one time where x^{near} satisfies $\Sigma^0(x^{mult}, x^d) \leq \Sigma^0(x^{near}, x^d)$. x^{near} can be thought of as a point close to x^d because $\mu(x^{near})$ has a higher initial covariance with $\mu(x^d)$ than $\mu(x^{mult})$ does. We define $\beta_1 = \Sigma^0(x^{mult}, x^{near})$ and $\beta_2 = \Sigma^0(x^{near}, x^d)$. Note that $\beta_0 \leq \beta_2$ and $0 < \beta_0, \beta_1, \beta_2 \leq \beta$; Figure 5.1 visually shows the relationships between the points. The decrease in the conditional variance of $\mu(x^d)$ from measuring x^{near} is

$$Var^n[\mu(x^d)] - Var^{n+1}[\mu(x^d)] = \left(\beta_2 - \frac{n\beta_0\beta_1}{n\beta + \lambda}\right)^2 \left(\beta - \frac{n\beta_1^2}{n\beta + \lambda} + \lambda\right)^{-1}. \tag{5.3}$$

We want to show that if we have measured x^{mult} n times (and measured nothing else) that the amount we can lower the conditional variance of $\mu(x^d)$ by observing x^{mult} again given in (5.2) is smaller than the amount given in (5.3) if we observe a new point x^{near}. We verify this is true algebraically in Appendix 8.4. We have shown that, for any $n \geq 0$, if we have sampled the decisions $x^0, ..., x^{n-1} = x^{mult}$ the additional decrease in the conditional variance of $\mu(x^d)$ would be smallest by setting $x^n = x^{mult}$. This is true for $n = 0, 1, 2, ...$, so using an induction argument this proves that $max_{x_0,...,x_{n-1} \in B(x^{acc},\epsilon)} Var^n[\mu(x^d)]$ equals $Var^n[\mu(x^d)]$ under π^{mult}. As explained above, $max_{x_0,...,x_{n-1} \in B(x^{acc},\epsilon)} Var^n[\mu(x^d)]$ is an upper bound on the conditional variance of $\mu(x^d)$ after we have measured n points in $B(x^{acc}, \epsilon)$ (and possibly more points outside $B(x^{acc}, \epsilon)$). Under π^{mult}, $Var^n[\mu(x^d)] = \beta - (\Sigma^0(x^{mult}, x^d))^2 \frac{n}{\beta n + \lambda}$ which gives us the upper bound.□

PROPOSITION 5.3. *Let x^{acc} be an accumulation point of the sequence of sampling decisions $\{x^n\}_{n=0}^{\infty}$. Consider a point $x^d \in B(x^{acc}, \epsilon)$ using the Euclidean distance. Then $\lim_{n\to\infty} Var^n[\mu(x^d)] \leq \beta - \beta \exp(-8 \sum_{i=1}^{p} \alpha_i \epsilon^2)$ for every sample path.*

Proof: We first show that $Var^n[\mu(x^d)]$ converges because it is a decreasing sequence that is bounded below by zero. If we measure x^n at time n, the equation for the conditional variance becomes

$$\Sigma^{n+1}(x^d, x^d) = \Sigma^n(x^d, x^d) - (\Sigma^n(x^n, x^d))^2 (\Sigma^n(x^n, x^n) + \lambda)^{-1}. \tag{5.4}$$

The second term in (5.4) is clearly positive and thus $\Sigma^{n+1}(x^d, x^d) \leq \Sigma^n(x^d, x^d)$. Now, n is arbitrary, so we can conclude that $Var^n(\mu(x^d))$ is a decreasing sequence bounded below by zero. We define $Var^{\infty}[\mu(x^d)]$ as the limit of $Var^n[\mu(x^d)]$.

x^{acc} is an accumulation point so for all $\epsilon > 0$ there are an infinite number of n with $x^n \in B(x^{acc}, \epsilon)$. We now put an upper bound on $Var^n[\mu(x^d)]$. Under the policy π^{mult} of only measuring x^{mult} we can see

$$\lim_{n\to\infty} Var^{\pi^{mult},n}[\mu(x)] = \beta - \frac{(\Sigma^0(x^{mult}, x))^2}{\beta}.$$

Let $\{k_n\}_{n=0}^{\infty}$ be a subsequence of natural numbers such that the policy π chooses $x^{k_n} \in B(x^{acc}, \epsilon)\ \forall n$. Let x^{mult} satisfy $\Sigma^0(x^{mult}, x^d) \leq \Sigma^0(x, x^d), \forall x \in B(x^{acc}, \epsilon)$. Using Proposition 5.2, we see that

$$Var^{\pi,k_n}[\mu(x^d)] \leq Var^{\pi^{mult},n}[\mu(x^d)] = \beta - (\Sigma^0(x^{mult}, x^d))^2 \frac{n}{\beta n + \lambda_0}. \quad (5.5)$$

Now, letting n go to infinity we get

$$Var^{\infty}[\mu(x^d)] = \lim_{n\to\infty} Var^{\pi,n}[\mu(x^d)] = \lim_{n\to\infty} Var^{\pi,k_n}[\mu(x^d)] \leq \beta - \frac{(\Sigma^0(x^{mult}, x^d))^2}{\beta}. \quad (5.6)$$

This equation holds for any x^{mult} which satisfies $\Sigma^0(x^{mult}, x^{acc}) \leq \Sigma^0(x, x^{acc}), \forall x \in B(x^{acc}, \epsilon)$ for a fixed $\epsilon > 0$. We next take the supremum over all such x^{mult} to obtain

$$\begin{aligned} Var^{\infty}[\mu(x^d)] &\leq \sup_{x\in B(x^{acc},\epsilon)} \left(\beta - \frac{(\Sigma^0(x, x^d))^2}{\beta}\right) \\ &= \beta - \frac{(\inf_{x\in B(x^{acc},\epsilon)}(\Sigma^0(x, x^d))^2}{\beta} \\ &\leq \beta - \frac{\inf_{x\in B(x^{acc},\epsilon)}(\beta e^{-\sum_{i=1}^p \alpha_i (x_i - x_i^d)^2})^2}{\beta} \\ &\leq \beta - \frac{(\beta e^{-\sum_{i=1}^p \alpha_i 4\epsilon^2})^2}{\beta} = \beta - \beta e^{-8\sum_{i=1}^p \alpha_i \epsilon^2}. \end{aligned} \quad (5.7)$$

Equation (5.7) uses the fact that $(x_i - x_i^d)^2 \leq 4\epsilon^2$ because $x, x^d \in B(x^{acc}, \epsilon)$ using the Euclidean distance. □

COROLLARY 5.4. *Since Proposition 5.3 was true for an arbitrary $\epsilon > 0$ and $\lim_{\epsilon\to 0} \beta - \beta e^{-8\sum_{i=1}^p \alpha_i \epsilon^2} = 0$, we can conclude that $\lim_{n\to\infty} Var^n[\mu(x^{acc})] = 0$.*

We now want to show that the knowledge gradient for continuous parameters of the points being sampled as n goes to infinity gets arbitrarily close to zero.

THEOREM 5.5. *Using the KGCP policy, $\liminf_{n\to\infty} \sup_{x\in\mathcal{X}} \bar{\nu}^{KG,n}(x) = 0$ for every sample path.*

Proof: Using equation 5.1 from the proof of Proposition 5.1, we put an upper bound on the knowledge gradient for continuous parameters at x^n,

$$\bar{\nu}^{KG,n}(x^n) \leq \frac{2}{\sqrt{2\pi}} \sqrt{\frac{\beta Var^n[\mu(x^n)]}{\lambda}}. \quad (5.8)$$

First, the sequence of sampling decisions is a bounded sequence in $\mathbb{R}^p$ and thus has an accumulation point, x^{acc}. Also, the sequence $\{\sup_{x\in\mathcal{X}} \bar{\nu}^{\mathrm{KG},n}(x)\}_{n=0}^{\infty}$ is a non-negative sequence because the knowledge gradient for continuous parameters is non-negative. Let $\{k_n\}_{n=0}^{\infty}$ be a subsequence of natural numbers such that the KGCP policy chooses $x^{k_n} \in B(x^{acc}, \epsilon)\ \forall n$. Now using Proposition 5.3 we write $\lim_{n\to\infty} Var^n[\mu(x^{k_n})] \leq \beta - \beta e^{-8\sum_{i=1}^p \alpha_i \epsilon^2}$. Combining this with (5.8) we get,

$$0 \leq \liminf_{n\to\infty} \bar{\nu}^{KG,k_n}(x^{k_n}) \leq \liminf_{n\to\infty} \frac{2}{\sqrt{2\pi}} \sqrt{\frac{\beta Var^{k_n}[\mu(x^{k_n})]}{\lambda}} \leq \frac{2}{\sqrt{2\pi}} \sqrt{\frac{\beta(\beta - \beta e^{-8\sum_{i=1}^p \alpha_i \epsilon^2})}{\lambda}}.$$

Since this equation was true for an arbitrary $\epsilon > 0$ and $\lim_{\epsilon \to 0} \frac{2}{\sqrt{2\pi}} \sqrt{\frac{\beta(\beta - \beta e^{-8 \sum_{i=1}^{p} \alpha_i \epsilon^2})}{\lambda}} = 0$, we can conclude that $\liminf_{n \to \infty} \bar{\nu}^{KG,k_n}(x^{k_n}) = 0$. This implies that $\liminf_{n \to \infty} \bar{\nu}^{KG,n}(x^n) = 0$ as well because the lim inf of a sequence is less than or equal to the lim inf of one of its subsequences. Recalling that under the KGCP policy $\bar{\nu}^{KG,n}(x^n) = \sup_{x \in \mathcal{X}} \bar{\nu}^{\mathrm{KG},n}(x)$ by Assumption 5.0.4 and because $\bar{\nu}^{\mathrm{KG},n}(x)$ is continuous and $\mathcal{X}$ is compact, we arrive at the desired result. □

For the following theorems we need Assumption 5.0.2 that prevents the updated mean from approaching infinity or negative infinity. We need Assumption 5.0.3 which ensures the function does not become perfectly correlated at two different decisions; this seems intuitive but is not trivial to prove.

THEOREM 5.6. *If Assumptions 5.0.1, 5.0.2, 5.0.3, and 5.0.4 are satisfied and if* $\liminf_{n \to \infty} \sup_{x \in \mathcal{X}} \bar{\nu}^{KG,n}(x) = 0$, *then* $Var^n(\mu(x))$ *converges to zero for all* x.

Proof:

$$
\begin{aligned}
&\bar{\nu}^{KG,n}(x) \\
&= \mathbb{E}\left[\max_{i=0,..,n} \mu^{n+1}(x^i) | \mathcal{F}^n, x^n = x\right] - \max_{i=0,..,n} \mu^n(x^i) |^{x^n = x} \\
&= \mathbb{E}\left[\max_{i=0,..,n} \mu^{n+1}(x^i) | \mathcal{F}^n, x^n = x\right] - \max(\mu^n(x^{i^\star}), \mu^n(x)) \qquad (5.9) \\
&\geq \mathbb{E}\left[\max(\mu^{n+1}(x^{i^\star}), \mu^{n+1}(x)) | \mathcal{F}^n\right] - \max(\mu^n(x^{i^\star}), \mu^n(x)) \\
&= \mathbb{E}\left[\max\left(\mu^n(x^{i^\star}) + \tilde{\sigma}_{i^\star}(\bar{\Sigma}^n, x) Z^{n+1}, \mu^n(x) + \tilde{\sigma}_n(\bar{\Sigma}^n, x) Z^{n+1}\right) | \mathcal{F}^n\right] - \max(\mu^n(x^{i^\star}), \mu^n(x)) \\
&= \mathbb{E}\left[\max\left(a_1 + b_1 Z^{n+1}, a_2 + b_2 Z^{n+1}\right)\right] - \max(a_1, a_2) \qquad (5.10) \\
&= \begin{cases} \int_{-\infty}^{\frac{a_2 - a_1}{b_1 - b_2}} (a_2 + b_2 z) f(z) dz + \int_{\frac{a_2 - a_1}{b_1 - b_2}}^{\infty} (a_1 + b_1 z) f(z) dz - \max(a_1, a_2), & \text{if } b_2 \leq b_1 \\ \int_{-\infty}^{\frac{a_2 - a_1}{b_1 - b_2}} (a_1 + b_1 z) f(z) dz + \int_{\frac{a_2 - a_1}{b_1 - b_2}}^{\infty} (a_2 + b_2 z) f(z) dz - \max(a_1, a_2), & \text{if } b_1 < b_2 \end{cases} \\
&= \begin{cases} a_2 \Phi(\frac{a_2 - a_1}{b_1 - b_2}) - b_2 \phi(\frac{a_2 - a_1}{b_1 - b_2}) + a_1 (1 - \Phi(\frac{a_2 - a_1}{b_1 - b_2})) + b_1 \phi(\frac{a_2 - a_1}{b_1 - b_2}) - \max(a_1, a_2), & \text{if } b_2 \leq b_1 \\ a_1 \Phi(\frac{a_2 - a_1}{b_1 - b_2}) - b_1 \phi(\frac{a_2 - a_1}{b_1 - b_2}) + a_2 (1 - \Phi(\frac{a_2 - a_1}{b_1 - b_2})) + b_2 \phi(\frac{a_2 - a_1}{b_1 - b_2}) - \max(a_1, a_2), & \text{if } b_1 < b_2 \end{cases} \\
&= a_2 \Phi\left(\frac{a_2 - a_1}{|b_1 - b_2|}\right) + a_1 \left(1 - \Phi\left(\frac{a_2 - a_1}{|b_1 - b_2|}\right)\right) + |b_1 - b_2| \phi\left(\frac{a_2 - a_1}{|b_1 - b_2|}\right) - \max(a_1, a_2) \\
&= -|a_2 - a_1| \Phi\left(\frac{-|a_2 - a_1|}{|b_1 - b_2|}\right) + |b_1 - b_2| \phi\left(\frac{|a_2 - a_1|}{|b_1 - b_2|}\right). \qquad (5.11)
\end{aligned}
$$

In (5.9), we define $i^\star = \arg\max_{i=0,..,n-1} \mu^n(x^i)$. In (5.10), for convenience, we define $a_1 = \mu^n(x^{i^\star})$, $a_2 = \tilde{\sigma}_{i^\star}(\bar{\Sigma}^n, x)$, $b_1 = \mu^n(x)$, and $b_2 = \tilde{\sigma}_n(\bar{\Sigma}^n, x)$. The term in (5.11) is nonnegative and decreases as $|a_2 - a_1|$ increases or $|b_1 - b_2|$ decreases. Equation (5.11) holds for all decisions x. Now, assume there is a decision x^{b_1} such that $\lim_{n \to \infty} Var^n[\mu(x^{b_1})] = \epsilon_1 > 0$. This limit exists because $Var^n[\mu(x^{b_1})]$ is a decreasing sequence bounded below by zero as shown in (5.4). Then (5.11) becomes

$$
\begin{aligned}
\bar{\nu}^{KG,n}(x^{b_1}) \geq &-|\mu^n(x^{b_1}) - \mu^n(x^{i^\star})| \Phi\left(\frac{-|\mu^n(x^{b_1}) - \mu^n(x^{i^\star})|}{|\tilde{\sigma}_{i^\star}(\bar{\Sigma}^n, x^{b_1}) - \tilde{\sigma}_n(\bar{\Sigma}^n, x^{b_1})|}\right) \\
&+ |\tilde{\sigma}_{i^\star}(\bar{\Sigma}^n, x^{b_1}) - \tilde{\sigma}_n(\bar{\Sigma}^n, x^{b_1})| \phi\left(\frac{|\mu^n(x^{b_1}) - \mu^n(x^{i^\star})|}{|\tilde{\sigma}_{i^\star}(\bar{\Sigma}^n, x^{b_1}) - \tilde{\sigma}_n(\bar{\Sigma}^n, x^{b_1})|}\right). \qquad (5.12)
\end{aligned}
$$

Now by assumptions 5.0.2 and 5.0.3, $\exists c_1, c_2$ such that

$$
\begin{aligned}
&\limsup_{n \to \infty} |\mu^n(x) - \mu^n(x^{i^\star})| \leq c_1 < \infty, \\
&\limsup_{n \to \infty} Corr^n[\mu(x^{b_1}), \mu(x^{i^\star})] \leq c_2 < 1.
\end{aligned}
$$

We can now put a lower bound on $|\tilde{\sigma}_{i^\star}(\bar{\Sigma}^n, x^{b_1}) - \tilde{\sigma}_n(\bar{\Sigma}^n, x^{b_1})|$.

$$
\begin{aligned}
&| \ \tilde{\sigma}_{i^\star}(\bar{\Sigma}^n, x^{b_1}) - \tilde{\sigma}_n(\bar{\Sigma}^n, x^{b_1})| \\
&= \frac{|Var^n[\mu(x^{b_1})] - Cov^n[\mu(x^{b_1}), \mu(x^{i^\star})]|}{\lambda + Var^n[\mu(x^{b_1})]} \\
&\geq \frac{Var^n[\mu(x^{b_1})] - Corr^n[\mu(x^{b_1}), \mu(x^{i^\star})]\sqrt{Var^n[\mu(x^{b_1})]Var^n[\mu(x^{i^\star})]}}{\lambda + \beta} \\
&\geq \frac{(1 - Corr^n[\mu(x^{b_1}), \mu(x^{i^\star})])\epsilon_1}{\lambda + \beta}.
\end{aligned}
$$

And now taking the limit inferior, we get

$$
\begin{aligned}
\liminf_{n\to\infty} |\tilde{\sigma}_{i^\star}(\bar{\Sigma}^n, x^{b_1}) - \tilde{\sigma}_n(\bar{\Sigma}^n, x^{b_1})| &\geq \liminf_{n\to\infty} \frac{(1 - Corr^n[\mu(x^{b_1}), \mu(x^{i^\star})])\epsilon_1}{\lambda + \beta} \\
&\geq \frac{c_2\epsilon_1}{\lambda + \beta} \\
&= c_3 > 0.
\end{aligned}
$$

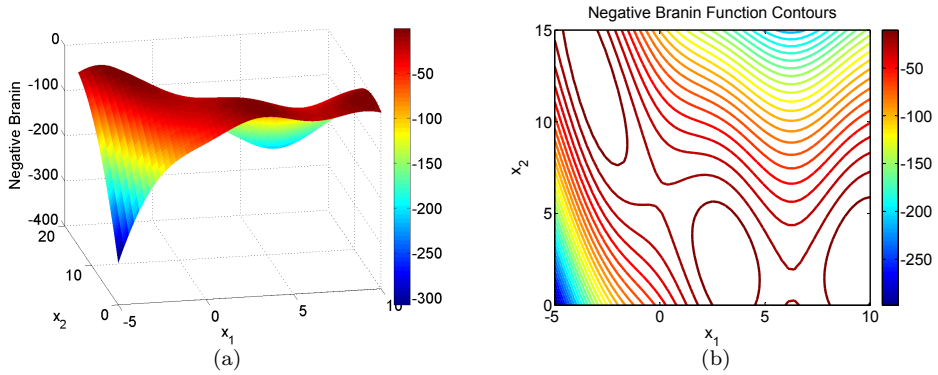
Going back to (5.12) and taking the limit inferior, we can now write

$$
\liminf_{n\to\infty} \bar{\nu}^{KG,n}(x^{b_1}) \geq -c_1\Phi\left(\frac{-c_1}{c_3}\right) + c_3\phi\left(\frac{c_1}{c_3}\right) > 0. \tag{5.13}
$$

By assumption the limit inferior of the supremum of the knowledge gradient for continuous parameters over all decisions is zero and thus (5.13) provides a contradiction. □

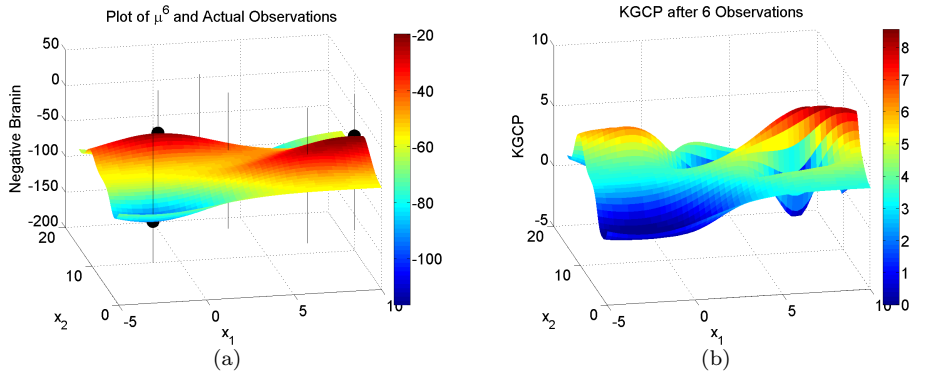
COROLLARY 5.7. *Under the KGCP Policy, if Assumptions 5.0.1, 5.0.2, 5.0.3, and 5.0.4 are satisfied, then* $\lim_{n\to\infty} Var^n[\mu(x)] = 0$ *for all* x.
Proof: Combining Theorem 5.5 and Theorem 5.6 we are left with the desired result. □

6. Numerical Results. In this section we give an illustrative example of the KGCP policy as well as analyzing its performance on several standard test functions. We first illustrate the KGCP policy on the 2-dimensional Branin function and set the variance of the normally distributed observation noise to one ($\lambda = 1$). We plot the true Branin function in Figure 6.1. We stick with the more conservative convention of an initial LHS design using two times the number of dimensions plus two ($2p+2$) used in [10] ([22] suggests using $10p$). After every observation we estimate the parameters (α, β, λ, and μ^0) with maximum likelihood estimation. Our estimate of the function after the initial 6 observations is shown in Figure 6.2(a), and the knowledge gradient for continuous parameters for each decision is shown in Figure 6.2(b). The knowledge gradient for continuous parameters is higher at decisions that have higher estimates or more uncertainty or both. At this point, after each observation, we update our estimate of the parameters and then choose our sampling decision by maximizing the knowledge gradient for continuous parameters. We repeat this several times, and Figure 6.3 shows the estimate of the function after 20 total observations chosen with the KGCP policy. Comparing these estimates with the true function shown in Figure 6.1, we visually see that the policy has done a good job estimating the upper regions of the function as desired.



(a) (b)

FIG. 6.1. *(a) The negative of the Branin function. (b) A contour plot of the negative Branin function. We will maximize the negative of the Branin function using noisy observations normally distributed around the true function.*



(a) (b)

FIG. 6.2. *(a) The estimate of the function after 6 observations. The actual observations are plotted as well. (b) The knowledge gradient for continuous parameters surface is plotted. The height is a measure of how much we expect the maximum of the estimate of the function to increase by measuring the corresponding decision. We choose the next sampling decision by finding the decision which maximizes the knowledge gradient for continuous parameters shown in 6.2(b).*

6.1. Standard Test Functions. Next we compare the KGCP policy with sequential kriging optimization (SKO) from [15] on expensive functions with observation noise. We use the various test functions used in [10], [16], and [15] as the true mean and add on normally distributed observation noise with variance λ. We define the opportunity cost as,

$$OC = \max_i \mu(i) - \mu(i^\star), \tag{6.1}$$

where $i^\star = \arg\max_i \mu^n(i)$, and Table 6.1 shows the performance on the different functions. These functions were designed to be minimized so the KGCP policy was applied to the negative of the functions. Each policy was run 500 times with the specified amount of observation noise. Table 6.1 gives the sample mean and sample standard deviation of the mean of the opportunity cost after 50 iterations for each policy. (To get the sample standard deviation of the opportunity cost you would multiply by $\sqrt{500}$). The means of the opportunity costs which are significantly better (using Welch's t test at the .05 level (see [44])) are bolded. The results are given

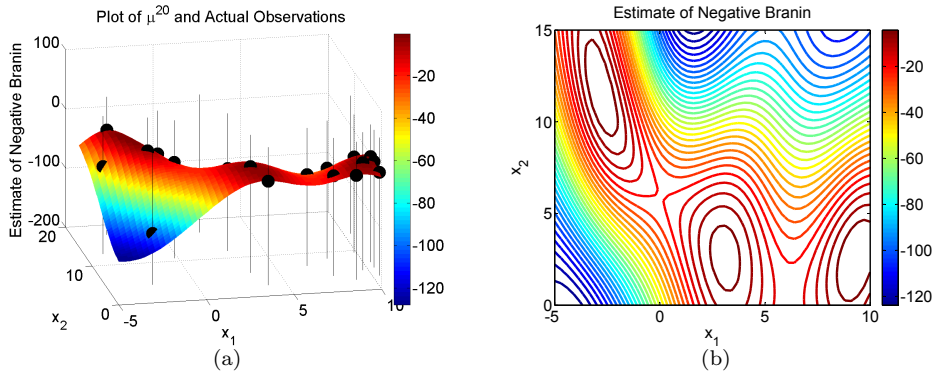


FIG. 6.3. *(a) The estimate of the function after 20 observations. The actual observations are plotted as well. (b) The contour plot of the estimate of the function after 20 observations.*

for different levels of noise; λ is the variance of the normally distributed noise in the observations. Because a Gaussian process (GP) is only an approximation (a surrogate) for the preceding test functions, we next apply KGCP and SKO to functions that are guaranteed to be GP's. Each GP row of Table 6.1 summarizes the results of running the policies on 500 GP's created as follows: a function was generated from a 1-dimensional GP with the specified parameters of the covariance matrix in (3.2) over a 300 point grid on the interval $[0, 15]$. The standard deviation of each function, σ, is given as well to give a frame of reference for the values of λ. This number was created by taking the standard deviation of function values over a discretized grid. For all these runs (even the Gaussian process surfaces) an initial LHS design of $2p+2$ function evaluations is used and maximum likelihood estimation is performed after each iteration to update the estimates of α, β, λ, and μ^0 (see [30]).

KGCP and SKO appear to have similar performance on Hartman 3 and Six Hump Camelback test functions. However, the KGCP policy does significantly better on the Ackley 5 and Branin test functions, as well as most of the Gaussian process functions. To get an idea of the rate of convergence of the KGCP policy, we plot the performance on the Gaussian processes in Figure 6.4. These promising simulations demonstrate that the KGCP algorithm is a very competitive policy.

7. Conclusion and Future Work. The knowledge gradient for continuous parameters is applicable to problems with continuous decision variables and observation noise and is similar to the expected improvement used in EGO when there is no observation noise. We presented a gradient ascent algorithm to approximately maximize the knowledge gradient for continuous parameters. The KGCP policy is very competitive with SKO and has nice convergence theory, giving conditions under which our uncertainty about the maximum of the expensive function with observation noise disappears. Extensions could include additional research with a priori distributions as well as additional approximations to speed up computations as the number of observations get large. Additional issues for further investigation are evaluating the algorithm on problems with larger dimensions, p, and extending the algorithm to unequal variances in the observation noise.

REFERENCES

		KGCP			SKO		
Test Function	$\sqrt{\lambda}$	$\mathbb{E}(OC)$	$\sigma(OC)$	Med	$\mathbb{E}(OC)$	$\sigma(OC)$	Med
Ackley 5 ($\mathcal{X} = [-15, 30]^5$)	$\sqrt{.1}$	**5.7304**	.1874	4.0964	7.8130	.1802	6.4978
	$\sqrt{1.0}$	**10.8315**	.2413	10.5855	12.6346	.2088	13.3955
$p = 5, \sigma = 1.126$	$\sqrt{10.0}$	**17.3670**	.1477	18.3281	18.1126	.1156	18.6481
Branin	$\sqrt{.1}$	**.0141**	.0044	.0046	.0460	.0023	.0302
	$\sqrt{1.0}$	**.0462**	.0039	.0234	.1284	.0218	.0737
$p = 2, \sigma = 51.885$	$\sqrt{10.0}$	**.2827**	.0186	.1386	.4396	.0248	.2685
Hartman3	$\sqrt{.1}$	**.0690**	.0063	.0249	.1079	.0075	.0650
	$\sqrt{1.0}$	.5336	.0296	.2658	.5012	.0216	.3737
$p = 3, \sigma = .938$	$\sqrt{10.0}$	1.8200	.0541	1.6182	1.8370	.0510	1.6552
Six Hump Camelback	$\sqrt{.1}$	**.0714**	.0087	.0698	.1112	.0059	.0797
	$\sqrt{1.0}$	.3208	.0192	.1315	.3597	.0156	.2035
$p = 2, \sigma = 3.181$	$\sqrt{10.0}$	1.0264	.0391	.8641	**.8488**	.0370	.6585
GP ($\alpha = .1, \beta = 100$)	$\sqrt{.1}$	.0076	.0057	.0000	.0195	.0041	.0043
	$\sqrt{1.0}$	.0454	.0243	.0018	.0888	.0226	.0182
$p = 1, \sigma = 8.417$	$\sqrt{10.0}$	.3518	.0587	.0337	.2426	.0216	.0535
GP ($\alpha = 1, \beta = 100$)	$\sqrt{.1}$	**.0077**	.0022	.0000	.0765	.0311	.0000
	$\sqrt{1.0}$	**.0270**	.0045	.0000	.1993	.0486	.0255
$p = 1, \sigma = 9.909$	$\sqrt{10.0}$	**.4605**	.1028	.0489	.6225	.0669	.1558
GP ($\alpha = 10, \beta = 100$)	$\sqrt{.1}$	**.1074**	.0259	.0000	.5302	.0799	.0000
	$\sqrt{1.0}$	**.1846**	.0286	.0000	.6638	.0839	.0839
$p = 1, \sigma = 10.269$	$\sqrt{10.0}$	**1.0239**	.1021	.1415	1.8273	.1450	.6290

TABLE 6.1

Performance on Standard Test Functions. Each row summarizes 500 runs of each policy on the specified test function with the specified observation noise variance. We define $\sigma(OC)$ as $Std(\mathbb{E}(OC))$ and Med as the median OC.

[1] A.J. BOOKER, J.E. DENNIS, P.D. FRANK, D.B. SERAFINI, V. TORCZON, AND M.W. TROSSET, *A rigorous framework for optimization of expensive functions by surrogates*, Structural and Multidisciplinary Optimization, 17 (1999), pp. 1–13.

[2] G.E. BOX AND N.R. DRAPER, *A basis for the selection of a response surface design*, Journal of the American Statistical Association, 54 (1959), pp. 622–654.

[3] A.R. CONN AND K. SCHEINBERG, *Geometry of sample sets in derivative-free optimization: polynomial regression and underdetermined interpolation*, IMA journal of numerical analysis, 28 (2008), p. 721.

[4] A.R. CONN, K. SCHEINBERG, AND P.L. TOINT, *Recent progress in unconstrained nonlinear optimization without derivatives*, Mathematical Programming, 79 (1997), pp. 397–414.

[5] N. CRESSIE, *The origins of kriging*, Mathematical Geology, 22 (1990), pp. 239–252.

[6] G. DENG AND M.C. FERRIS, *Adaptation of the UOBYQA algorithm for noisy functions*, Precedings of the 2006 Winter Simulation Conference, (2006), pp. 312–319.

[7] A.I.J. FORRESTER, A. SÓBESTER, AND A.J. KEANE, *Multi-fidelity optimization via surrogate modelling*, Proceeding of the Royal Society A, 463 (2007), pp. 3251–3269.

[8] A.I.J. FORRESTER, A. SOBESTER, AND A.J. KEANE, *Engineering Design via Surrogate Modelling: A Practical Guide*, John Wiley & Sons, Ltd., 2008.

[9] P. FRAZIER, W.B. POWELL, AND S. DAYANIK, *A knowledge-gradient policy for sequential information collection*, SIAM Journal on Control and Optimization, 47 (2008), pp. 2410–2439.

[10] P. FRAZIER, W.B. POWELL, AND S. DAYANIK, *The knowledge gradient policy for correlated normal beliefs*, INFORMS Journal on Computing, 21 (2009), pp. 599–613.

[11] M.C. FU, F.W. GLOVER, AND J. APRIL, *Simulation optimization: A review, new developments, and applications*, Proceedings of the 2005 Winter Simulation Conference, (2005), pp. 83–95.

[12] P. GLASSERMAN, *Gradient Estimation via Perturbation Analysis*, Kluwer Academic Publishers, Norwell, Massachusetts, 1991.

[13] H.M. GUTMANN, *A radial basis function method for global optimization*, Journal of Global Optimization, 19 (2001), pp. 201–227.

[14] W.J. HILL AND W.G. HUNTER, *A review of response surface methodology: A literature survey*, Technometrics, 8 (1966), pp. 571–590.

[15] D. HUANG, T.T. ALLEN, W.I. NOTZ, AND N. ZENG, *Global optimization of stochastic black-box systems via sequential kriging meta-models,*, Journal of Global Optimization, 34 (2006), pp. 441–446.

[16] D.R. JONES, M. SCHONLAU, AND W.J. WELCH, *Efficient global optimization of expensive black-*

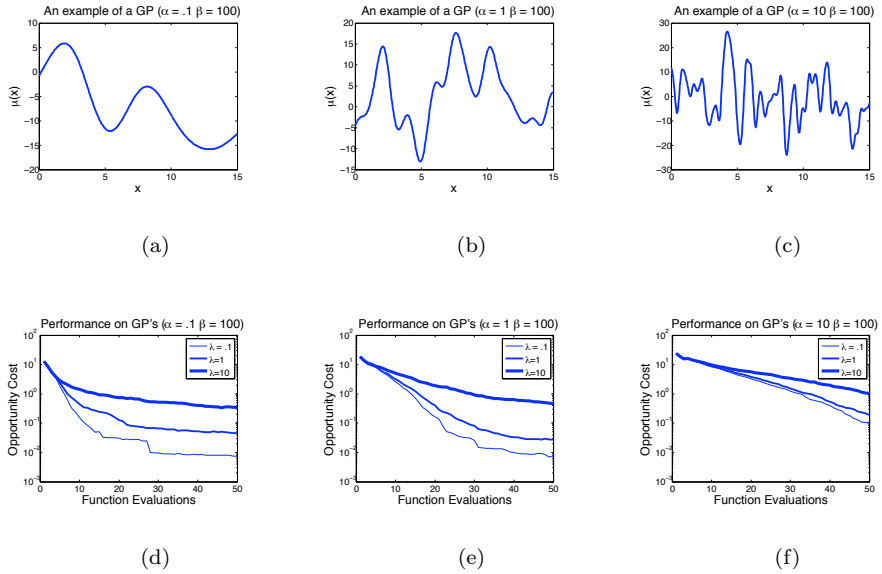


FIG. 6.4. *(a)-(c) show examples of Gaussian Processes with the given covariance parameters. (d)-(f) show the mean opportunity cost of the KGCP policy on the various Gaussian processes.*

box functions, Journal of Global Optimization, 13 (1998), pp. 455–492.

[17] J. KIEFER AND J. WOLFOWITZ, *Stochastic estimation of the maximum of a regression function*, Annals Mathematical Statistics, 23 (1952), pp. 462–466.

[18] J.P.C. KLEIJNEN, *Kriging metamodeling in simulation: A review*, European Journal of Operations Research, 192 (2009), pp. 707–716.

[19] J.P.C. KLEIJNEN, W. BEERS, AND I. NIEUWENHUYSE, *Constrained optimization in expensive simulation; novel approach*, European Journal of Operational Research, 202 (2010), pp. 164–174.

[20] H. 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 (1964), pp. 97–106.

[21] M. LOCATELLI, *Bayesian algorithms for one-dimensional global optimization*, Journal of Global Optimization, 10 (1997), pp. 57–76.

[22] J.L. LOEPPKY, J. SACKS, AND W.J. WELCH, *Choosing the sample size of a computer experiment: A practical guide*, Technometrics, 51 (2009), pp. 366–376.

[23] G. MATHERON, *Principles of geostatistics*, Economic Geology, 58 (1963), pp. 1246–1266.

[24] R.J. MEINHOLD AND N.D. SINGPURWALLA, *Understanding the Kalman Filter*, The American Statistician, 37 (1983), pp. 123–127.

[25] J. MOCKUS, *Application of bayesian approach to numerical methods of global and stochastic optimization*, Journal of Global Optimization, 4 (1993), pp. 347–365.

[26] R.H. MYERS, A.I. KHURI, AND W.H. CARTER JR, *Response surface methodology: 1966-1988*, Technometrics, 31 (1989), pp. 137–157.

[27] R.H. MYERS, A.I. KHURI, AND G. VINING, *Response surface alternatives to the taguchi robust parameter design approach*, The American Statistician, 46 (1992), pp. 131–139.

[28] R.H. MYERS, D.C. MONTGOMERY, AND C.M ANDERSON-COOK, *Response Surface Methodology*, Wiley, 2009.

[29] M.J.D. POWELL, *UOBYQA: Unconstrained Optimization by Quadratic Approximation*, Mathematical Programming, 92 (2002).

[30] C.E. RASMUSSEN AND C.K.I. WILLIAMS, *Gaussian Processes for Machine Learning*, MIT Press, 2006.

[31] R.G. REGIS AND C.A. SHOEMAKER, *Constrained global optimization of expensive black box functions using radial basis functions*, Journal of Global Optimization, 31 (2005), pp. 153–171.

[32] H. ROBBINS AND S. MONRO, *A stochastic approximation method*, Annals of Math. Stat., 22

(1951), pp. 400–407.
[33] R.Y. Rubinstein and A. Shapiro, *Optimization of static simulation models by the score function method*, Mathematics and Computers in Simulation, 32 (1990), pp. 373–392.
[34] J. Sacks, W.J. Welch, T.J. Mitchell, and H.P. Wynn, *Design and analysis of computer experiments*, Statistical Science, 4 (1989), pp. 409–423.
[35] M.J. Sasena, *Flexibility and Efficency Enhancements for Constrained Global Design Optimization with Kriging Approximations*, PhD thesis, University of Michigan, 2002.
[36] M. Schonlau and W.J. Welch, *Global optimization with nonparametric function fitting*, Proc. Section on Physical and Engineering Sciences, American Statistical Association, (1996), pp. 183–186.
[37] M. Schonlau, W.J. Welch, and D.R. Jones, *Global versus local search in constrained optimization of computer models*, Lecture Notes-Monograph Series, 34 (1998), pp. 11–25.
[38] D.W. Scott, *Parametric statistical model by minimum integrated square error*, Technometrics, 43 (2001), pp. 274–285.
[39] J.C. Spall, *Introduction to Stochastic Search and Optimization*, Wiley-Interscience, 2003.
[40] A. Torn and A. Zilinskas, *Global Optimization, Lecture Notes in Computer Science*, Springer-Verlag, 1989.
[41] E. Vazquez and J. Bect, *Convergence properties of the expected improvement algorithm with fixed mean and covariance functions*, Journal of Statistical Planning and Inference, 140 (2010), pp. 3088–3095.
[42] E. Vazquez, J. Villemonteix, M. Sidorkiewicz, and E. Walter, *Global optimization based on noisy evaluations: an empiricial study of two statistical approaches*, Journal of Physics: Conference Series, 135 (2008).
[43] J. Villemonteix, E. Vazquez, and E. Walter, *An informational approach to the global optimization of expensive-to-evaluate functions*, Journal of Global Optimization, 44 (2009), pp. 509–534.
[44] B.L. Welch, *The generalization of 'student's' problem when several different population variances are involved*, Biometrika, 34 (1947), pp. pp. 28–35.
[45] A.G. Zhilinskas, *Single-step bayesian search method for an extremum of functions of a single variable*, Cybernetics and Systems Analysis, 11 (1975), pp. 160–166.

8. Online Appendix.

8.1. Computing $\nabla_{x^n}\mu^n(x^i)$. If $i < n$ then $\mu^n_{x^i}$ does not depend on x^n so $\nabla_{x^n}\mu^n(x^i) = 0$. Now consider when $i = n$. We start with equation (3.9) for $\mu^n(x^n)$ where x^n has not been sampled and then simplify.

$$\begin{aligned}\mu^n(x^n) &= \mu^0(x^n) + e_{n+1}^T\bar{\Sigma}^0\begin{bmatrix} I_n \\ - \\ \vec{0}^T\end{bmatrix}[S^n]^{-1}\tilde{y}^n \\ &= \mu^0(x^n) + \begin{bmatrix}\Sigma^0(x^0,x^n) & ,\cdots, & \Sigma^0(x^{n-1},x^n)\end{bmatrix}[S^n]^{-1}\tilde{y}^n\end{aligned}$$

Now, because $[S^n]^{-1}\tilde{y}^n$ does not depend on the decision x^n, we can easily take the gradient.

$$\begin{aligned}\nabla_{x^n}\mu^n(x^n) &= \nabla_{x^n}\mu^0(x^n) + \begin{bmatrix}\nabla_{x^n}\Sigma^0(x^0,x^n) & ,\cdots, & \nabla_{x^n}\Sigma^0(x^{n-1},x^n)\end{bmatrix}[S^n]^{-1}\tilde{y}^n \\ &= \nabla_{x^n}\mu^0(x^n) + J^n[S^n]^{-1}\tilde{y}^n. \qquad (8.1)\end{aligned}$$

where we J^n is defined in (4.15). When going from (4.15) to (4.16) we used the fact that the covariance function was of the form specified in (3.2).

8.2. Computing $\nabla_{x^n}\tilde{\sigma}_i(\Sigma^n, x^n)$. First, recall that

$$\tilde{\sigma}_i(\bar{\Sigma}^n, x^n) = \frac{e_{x^i}^T\bar{\Sigma}^n e_{x^n}}{\sqrt{\lambda(x^n) + e_{x^n}^T\bar{\Sigma}^n e_{x^n}}}, \qquad i = 0,...,n. \qquad (8.2)$$

After we derive the gradient of the numerator and denominator of this equation, we can find the gradient of (8.2) by using the quotient rule for differentiation.

$$\nabla_{x^n}\tilde{\sigma}_i(\bar{\Sigma}^n, x^n) = \frac{\sqrt{\lambda(x^n) + e_{x^n}^T\bar{\Sigma}^n e_{x^n}}\nabla_{x^n}e_{x^i}^T\bar{\Sigma}^n e_{x^n} - e_{x^i}^T\bar{\Sigma}^n e_{x^n}\nabla_{x^n}\sqrt{\lambda(x^n) + e_{x^n}^T\bar{\Sigma}^n e_{x^n}}}{|\lambda(x^n) + e_{x^n}^T\bar{\Sigma}^n e_{x^n}|}$$

8.2.1. The Numerator. First we consider the numerator of (8.2).

$$\begin{aligned}e_{x^i}^T\bar{\Sigma}^n e_{x^n} &= e_{x^i}^T(I - \bar{K}^n\begin{bmatrix}I_n & | & \vec{0}\end{bmatrix})\bar{\Sigma}^0 e_{x^n} && (8.3)\\ &= e_{x^i}^T\bar{\Sigma}^0 e_{x^n} - e_{x^i}^T\bar{K}^n\begin{bmatrix}I_n & | & \vec{0}\end{bmatrix}\bar{\Sigma}^0 e_{x^n} && (8.4)\\ &= \Sigma^0(x^i,x^n) - e_{x^i}^T\bar{\Sigma}^0\begin{bmatrix} I_n \\ - \\ \vec{0}^T\end{bmatrix}[S^n]^{-1}\begin{bmatrix}I_n & | & \vec{0}\end{bmatrix}\bar{\Sigma}^0 e_{x^n} && (8.5)\\ &= \Sigma^0(x^i,x^n) - \begin{bmatrix}\Sigma^0(x^0,x^i) & ,\cdots, & \Sigma^0(x^{n-1},x^i)\end{bmatrix}[S^n]^{-1}\begin{bmatrix}\Sigma^0(x^0,x^n)\\ \vdots \\ \Sigma^0(x^{n-1},x^n)\end{bmatrix} && (8.6)\\ &= \Sigma^0(x^i,x^n) - \begin{bmatrix}\Sigma^0(x^0,x^n) & ,\cdots, & \Sigma^0(x^{n-1},x^n)\end{bmatrix}[S^n]^{-1}\begin{bmatrix}\Sigma^0(x^0,x^i)\\ \vdots \\ \Sigma^0(x^{n-1},x^i)\end{bmatrix} && (8.7)\end{aligned}$$

In (8.3) we used the definition of $\bar{\Sigma}^n$ in (3.10). From (8.4) to (8.6) we just inserted the definition of $\bar{K}^n$ given in (3.8). Going from (8.6) to (8.7) we took the transpose of the last term which is a scalar and used the fact that $[S^n]^{-1}$ is symmetric. We first

consider the case where $i < n$. In this case $[S^n]^{-1}\begin{bmatrix}\Sigma^0(x^0,x^i)\\ \vdots \\ \Sigma^0(x^{n-1},x^i)\end{bmatrix} = [S^n]^{-1}\Sigma^0 e_{x^i}$ and does not depend on x^n so we can easily compute the gradient,

$$\begin{aligned}\nabla_{x^n} e_{x^i}^T \bar{\Sigma}^n e_{x^n} &= \nabla_{x^n}\Sigma^0(x^i,x^n) - \left[\nabla_{x^n}\Sigma^0(x^0,x^n) \quad ,\cdots, \quad \nabla_{x^n}\Sigma^0(x^{n-1},x^n)\right][S^n]^{-1}\Sigma^0 e_{x^i} \\ &= 2DIAG(\alpha) * (x^i - x^n)\Sigma^0(x^i,x^n) - J^n[S^n]^{-1}\Sigma^0 e_{x^i}. \end{aligned} \tag{8.8}$$

Now we consider the case where $i = n$. Using standard matrix differentiation, we can compute the gradient.

$$\begin{aligned}\nabla_{x^n} e_{x^n}^T \bar{\Sigma}^n e_{x^n} &= \begin{bmatrix} 0 - 2\left[\frac{\partial}{\partial x_1^n}\Sigma^0(x^0,x^n) \quad ,\cdots, \quad \frac{\partial}{\partial x_1^n}\Sigma^0(x^{n-1},x^n)\right][S^n]^{-1}\begin{bmatrix}\Sigma^0(x^0,x^n)\\ \vdots \\ \Sigma^0(x^{n-1},x^n)\end{bmatrix} \\ \vdots \\ 0 - 2\left[\frac{\partial}{\partial x_p^n}\Sigma^0(x^0,x^n) \quad ,\cdots, \quad \frac{\partial}{\partial x_p^n}\Sigma^0(x^{n-1},x^n)\right][S^n]^{-1}\begin{bmatrix}\Sigma^0(x^0,x^n)\\ \vdots \\ \Sigma^0(x^{n-1},x^n)\end{bmatrix}\end{bmatrix} \\ &= -2\left[\nabla_{x^n}\Sigma^0(x^0,x^n) \quad ,\cdots, \quad \nabla_{x^n}\Sigma^0(x^{n-1},x^n)\right][S^n]^{-1}\begin{bmatrix}\Sigma^0(x^0,x^n)\\ \vdots \\ \Sigma^0(x^{n-1},x^n)\end{bmatrix} \\ &= -2J^n[S^n]^{-1}\begin{bmatrix}\Sigma^0(x^0,x^n)\\ \vdots \\ \Sigma^0(x^{n-1},x^n)\end{bmatrix}. \end{aligned}$$

8.2.2. The Denominator. Now we consider the denominator of (8.2).

$$\begin{aligned}&\sqrt{\lambda(x^n) + e_{x^n}^T \bar{\Sigma}^n e_{x^n}} \\ &= \sqrt{\lambda(x^n) + e_{x^n}^T (I - \bar{K}^n\left[I_n \quad | \quad \vec{0}\right])\bar{\Sigma}^0 e_{x^n}} &(8.9)\\ &= \sqrt{\lambda(x^n) + \Sigma^0(x^n,x^n) - e_{x^n}^T \bar{K}^n\left[I_n \quad | \quad \vec{0}\right]\bar{\Sigma}^0 e_{x^n}} &(8.10)\\ &= \sqrt{\lambda(x^n) + \Sigma^0(x^n,x^n) - e_{x^n}^T \bar{\Sigma}^0\begin{bmatrix}I_n \\ - \\ \vec{0}^T\end{bmatrix}[S^n]^{-1}\left[I_n \quad | \quad \vec{0}\right]\bar{\Sigma}^0 e_{x^n}} &(8.11)\\ &= \sqrt{\lambda(x^n) + \Sigma^0(x^n,x^n) - \left[\Sigma^0(x^0,x^n) \quad ,\cdots, \quad \Sigma^0(x^{n-1},x^n)\right][S^n]^{-1}\begin{bmatrix}\Sigma^0(x^0,x^n)\\ \vdots \\ \Sigma^0(x^{n-1},x^n)\end{bmatrix}} \end{aligned}$$

In (8.9) we inserted the definition of $\bar{\Sigma}^n$ given in (3.10). Going from (8.10) to (8.11) we inserted the definition of $\bar{K}^n$ given in (3.8). Now we take the gradient.

$$
\begin{aligned}
&\nabla_{x^n}\sqrt{\lambda(x^n)+e_{x^n}^T\Sigma^n e_{x^n}} \\
&= \begin{bmatrix} \frac{1}{2}(\lambda(x^n)+\Sigma^n(x^n,x^n))^{-\frac{1}{2}}(\frac{\partial}{\partial x_1^n}\lambda(x^n)-2\left[\frac{\partial}{\partial x_1^n}\Sigma^0(x^0,x^n) \quad ,\cdots, \quad \frac{\partial}{\partial x_1^n}\Sigma^0(x^{n-1},x^n)\right][S^n]^{-1}\begin{bmatrix}\Sigma^0(x^0,x^n)\\ \vdots \\ \Sigma^0(x^{n-1},x^n)\end{bmatrix}) \\ \vdots \\ \frac{1}{2}(\lambda(x^n)+\Sigma^n(x^n,x^n))^{-\frac{1}{2}}(\frac{\partial}{\partial x_p^n}\lambda(x^n)-2\left[\frac{\partial}{\partial x_p^n}\Sigma^0(x^0,x^n) \quad ,\cdots, \quad \frac{\partial}{\partial x_p^n}\Sigma^0(x^{n-1},x^n)\right][S^n]^{-1}\begin{bmatrix}\Sigma^0(x^0,x^n)\\ \vdots \\ \Sigma^0(x^{n-1},x^n)\end{bmatrix}) \end{bmatrix} \\
&= \frac{1}{2}(\lambda(x^n)+\Sigma^n(x^n,x^n))^{-\frac{1}{2}}(\nabla_{x^n}\lambda(x^n)-2\left[\nabla_{x^n}\Sigma^0(x^0,x^n) \quad ,\cdots, \quad \nabla_{x^n}\Sigma^0(x^{n-1},x^n)\right][S^n]^{-1}\begin{bmatrix}\Sigma^0(x^0,x^n)\\ \vdots \\ \Sigma^0(x^{n-1},x^n)\end{bmatrix}) \\
&= \frac{1}{2}(\lambda(x^n)+\Sigma^n(x^n,x^n))^{-\frac{1}{2}}\left(\nabla_{x^n}\lambda(x^n)-2J^n[S^n]^{-1}\begin{bmatrix}\Sigma^0(x^0,x^n)\\ \vdots \\ \Sigma^0(x^{n-1},x^n)\end{bmatrix}\right)
\end{aligned}
$$

8.3. Proof of Proposition 5.1. We derive the upper bound of the knowledge gradient for continuous parameters given in (5.1), starting with

$$
\begin{aligned}
&\mathbb{E}\left[\max_{i=0,..,n}\mu^{n+1}(x^i)\middle|\mathcal{F}^n,x^n=x\right] && (8.12)\\
&=\mathbb{E}\left[\max_{i=0,..,n}\mu^n(x^i)+\tilde{\sigma}_i(\bar{\Sigma}^n,x^n)Z^{n+1}\middle|\mathcal{F}^n,x^n=x\right] && (8.13)\\
&\le \max_{i=0,..,n}\mu^n(x^i)+\mathbb{E}\left[\max_{j=0,..,n}\tilde{\sigma}_j(\bar{\Sigma}^n,x^n)Z^{n+1}\middle|\mathcal{F}^n,x^n=x\right] \\
&= \max_{i=0,..,n}\mu^n(x^i)+\mathbb{E}\left[\max_{j=0,..,n}\tilde{\sigma}_j(\bar{\Sigma}^n,x^n)Z^{n+1}\mathbf{1}(Z^{n+1}>0)+\max_{k=0,..,n}\tilde{\sigma}_k(\bar{\Sigma}^n,x^n)Z^{n+1}\mathbf{1}(Z^{n+1}\le 0)\middle|\mathcal{F}^n,x^n=x\right] \\
&= \max_{i=0,..,n}\mu^n(x^i)+\mathbb{E}\left[Z^{n+1}\mathbf{1}(Z^{n+1}>0)\right]\max_{j=0,..,n}\tilde{\sigma}_j(\bar{\Sigma}^n,x^n)+\mathbb{E}\left[Z^{n+1}\mathbf{1}(Z^{n+1}\le 0)\right]\min_{k=0,..,n}\tilde{\sigma}_k(\bar{\Sigma}^n,x^n) \\
&= \max_{i=0,..,n}\mu^n(x^i)+\frac{1}{\sqrt{2\pi}}\max_{j=0,..,n}\tilde{\sigma}_j(\bar{\Sigma}^n,x^n)-\frac{1}{\sqrt{2\pi}}\min_{k=0,..,n}\tilde{\sigma}_k(\bar{\Sigma}^n,x^n) \\
&\le \max_{i=0,..,n}\mu^n(x^i)+\frac{2}{\sqrt{2\pi}}\max_{j=0,..,n}|\tilde{\sigma}_j(\bar{\Sigma}^n,x^n)|. && (8.14)
\end{aligned}
$$

We now need an upper bound on $|\tilde{\sigma}_j(\bar{\Sigma}^n,x^n)|$ in (8.14). We just note that

$$
\begin{aligned}
\left|\tilde{\sigma}_j(\bar{\Sigma}^n,x^n)\right| &= \left|\frac{e_{x^j}\bar{\Sigma}^n e_{x^n}}{\sqrt{\lambda+e_{x^n}^T\bar{\Sigma}^n e_{x^n}}}\right| \\
&= \left|\frac{Cov^n[\mu(x^j),\mu(x^n)]}{\sqrt{\lambda+Var^n[\mu(x^n)]}}\right| \\
&= \left|\frac{Corr^n[\mu(j),\mu(x^n)]\sqrt{Var^n[\mu(x^j)]Var^n[\mu(x^n)]}}{\sqrt{\lambda+Var^n[\mu(x^n)]}}\right| \\
&\le \left|\frac{\sqrt{Var^n[\mu(x^j)]Var^n[\mu(x^n)]}}{\sqrt{\lambda}}\right| \\
&= \sqrt{\frac{Var^n[\mu(x^j)]Var^n[\mu(x^n)]}{\lambda}}. && (8.15)
\end{aligned}
$$

Combining (8.14) and (8.15) we have an upper bound on the knowledge gradient for continuous parameters.

$$\bar{\nu}^{\mathrm{KG},n}(x) \leq \frac{2}{\sqrt{2\pi}} \max_{j=0,..,n} \sqrt{\frac{Var^n[\mu(x^j)]Var^n[\mu(x^n)]}{\lambda}} \leq \sqrt{\frac{2\beta Var^n[\mu(x^n)]}{\pi\lambda}} = \sqrt{\frac{2\beta Var^n[\mu(x)]}{\pi\lambda}} \tag{8.16}$$

The knowledge gradient for continuous parameters is nonnegative and the above upper bound on the knowledge gradient for continuous parameters of a decision x converges to zero as the conditional variance of $\mu(x)$ converges to zero.

8.4. Proof of Proposition 5.2. We derive how the conditional variance of $\mu(x^d)$ decreases if we repeatedly measure a particular point x^{mult} n times with noise variance λ for each observation. We define the policy π^{mult} which sets $x^0 = \cdots = x^{n-1} = x^{mult}$. Under this policy we see,

$$\begin{aligned}
&\Sigma^n(x,x) \\
&= e_x^T \bar{\Sigma}^n e_x \\
&= e_{x^n}^T (I - \bar{K}^n \begin{bmatrix} I_n & | & \vec{0} \end{bmatrix}) \bar{\Sigma}^0 e_{x^n} && (8.17) \\
&= \Sigma^0(x,x) - e_x^T \bar{K}^n \begin{bmatrix} I_n & | & \vec{0} \end{bmatrix} \bar{\Sigma}^0 e_x && (8.18) \\
&= \Sigma^0(x,x) - e_x^T \bar{\Sigma}^0 \begin{bmatrix} I_n \\ - \\ \vec{0}^T \end{bmatrix} [S^n]^{-1} \begin{bmatrix} I_n & | & \vec{0} \end{bmatrix} \bar{\Sigma}^0 e_x \\
&= \Sigma^0(x,x) - \begin{bmatrix} \Sigma^0(x^0,x) & ,\cdots, & \Sigma^0(x^{n-1},x) \end{bmatrix} [S^n]^{-1} \begin{bmatrix} \Sigma^0(x^0,x) \\ \vdots \\ \Sigma^0(x^{n-1},x) \end{bmatrix} && (8.19) \\
&= \Sigma^0(x,x) - \begin{bmatrix} \Sigma^0(x^0,x) & ,\cdots, & \Sigma^0(x^{n-1},x) \end{bmatrix} \left[\Sigma^0 + \lambda I_n\right]^{-1} \begin{bmatrix} \Sigma^0(x^0,x) \\ \vdots \\ \Sigma^0(x^{n-1},x) \end{bmatrix} \\
&= \beta - \begin{bmatrix} \Sigma^0(x^{mult},x) & ,\cdots, & \Sigma^0(x^{mult},x) \end{bmatrix} \left[\begin{bmatrix} \beta & \cdots & \beta \\ \vdots & \ddots & \vdots \\ \beta & \cdots & \beta \end{bmatrix} + \lambda I_n \right]^{-1} \begin{bmatrix} \Sigma^0(x^{mult},x) \\ \vdots \\ \Sigma^0(x^{mult},x) \end{bmatrix} \\
&= \beta - (\Sigma^0(x^{mult},x))^2 e^T \left[\beta \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix} + \lambda I_n \right]^{-1} e && (8.20) \\
&= \beta - (\Sigma^0(x^{mult},x))^2 \frac{n}{\beta n + \lambda}. && (8.21)
\end{aligned}$$

In (8.17) we insert the definition of $\bar{\Sigma}^n$ given in (3.10). In (8.18) we insert the definition of $\bar{K}^n$ given in (3.8). $[S^n]^{-1}$ is positive semi-definite, so the second term in (8.19) is nonnegative. In (8.20) e is a column vector of ones, and we simplify the expression

using the definition of the inverse of S^n,

$$[S^n]^{-1}\left[\beta\begin{bmatrix}1 & \cdots & 1\\ \vdots & \ddots & \vdots\\ 1 & \cdots & 1\end{bmatrix}+\lambda I_n\right]=I_n,$$

$$e^T[S^n]^{-1}\left[\beta\begin{bmatrix}1 & \cdots & 1\\ \vdots & \ddots & \vdots\\ 1 & \cdots & 1\end{bmatrix}+\lambda I_n\right]e=e^T I_n e,$$

$$e^T[S^n]^{-1}\left[\beta n e+\lambda e\right]=n,$$

$$e^T[S^n]^{-1}e=\frac{n}{\beta n+\lambda}. \tag{8.22}$$

First consider the change $Var^n(\mu(x^d))-Var^{n+1}(\mu(x^d))$ if we have measured x^{mult} n times and then measure x^{mult} one more time. We use (8.21) and assume $\Sigma^0(x,x)=\beta,\forall x$. Also, define $\beta_0=\Sigma^0(x^{mult},x^d)$. The decrease in the conditional variance of $\mu(x^d)$ from measuring x^{mult} once more is

$$\begin{aligned}
&Var^n(\mu(x^d))-Var^{n+1}(\mu(x^d))\\
&=\left(\beta-B_0^2 n(n\beta+\lambda)^{-1}\right)-\left(\beta-B_0^2(n+1)((n+1)\beta+\lambda)^{-1}\right) &(8.23)\\
&=B_0^2(n+1)((n+1)\beta+\lambda)^{-1}-B_0^2 n(n\beta+\lambda)^{-1}\\
&=\frac{B_0^2(n+1)(n\beta+\lambda)-B_0^2 n((n+1)\beta+\lambda)}{((n+1)\beta+\lambda)(n\beta+\lambda)}\\
&=\frac{\beta_0^2\lambda}{((n+1)\beta+\lambda)(n\beta+\lambda)}. &(8.24)
\end{aligned}$$

In (8.23) we just used (8.21) which gives an expression for $Var^n(\mu(x))$ if we measure x^{mult} n times and nothing else. Second we consider measuring the change in $Var^n(\hat{\mu}(x^d))-Var^{n+1}(\hat{\mu}(x^d))$ if we have measured x^{mult} n times and then measure x^{near} one time, where $x^{near}\in B(x^{acc},\epsilon)$ and satisfies $\Sigma^0(x^{mult},x^d)\leq\Sigma^0(x^{near},x^d)$. We define $\beta_1=\Sigma^0(x^{mult},x^{near})$ and $\beta_2=\Sigma^0(x^{near},x^d)$. Note that $\beta_0\leq\beta_2$ and $0<\beta_0,\beta_1,\beta_2\leq\beta$.

$$\begin{aligned}
&\Sigma^{n+1}(x^d,x^d)\\
&=\Sigma^n(x^d,x^d)-\Sigma^n(x^{near},x^d)\left(\Sigma^n(x^{near},x^{near})+\lambda\right)^{-1}\Sigma^n(x^{near},x^d) &(8.25)\\
&=\Sigma^n(x^d,x^d)-\left(\Sigma^n(x^{near},x^d)\right)^2\left(\Sigma^n(x^{near},x^{near})+\lambda\right)^{-1}\\
&=\Sigma^n(x^d,x^d)-\left(\Sigma^0(x^{near},x^d)-\frac{n\Sigma^0(x^{mult},x^d)\Sigma^0(x^{mult},x^{near})}{n\beta+\lambda}\right)^2\left(\Sigma^n(x^{near},x^{near})+\lambda\right)^{-1}\\
&=\Sigma^n(x^d,x^d)-\left(\beta_2-\frac{n\beta_0\beta_1}{n\beta+\lambda}\right)^2\left(\beta-(\Sigma^0(x^{mult},x^{near}))^2\frac{n}{n\beta+\lambda}+\lambda\right)^{-1} &(8.26)\\
&=\Sigma^n(x^d,x^d)-\left(\beta_2-\frac{n\beta_0\beta_1}{n\beta+\lambda}\right)^2\left(\beta-\frac{n\beta_1^2}{n\beta+\lambda}+\lambda\right)^{-1}
\end{aligned}$$

In (8.25) we use the recursive equation for updating the conditional variance. In (8.26) we plugged in the equation for $\Sigma^n(x^{near},x^d)$ which is derived in the same way as (8.21). Equivalently we can write

$$Var^n(\mu(x^d))-Var^{n+1}(\mu(x^d))=\left(\beta_2-\frac{n\beta_0\beta_1}{n\beta+\lambda}\right)^2\left(\beta-\frac{n\beta_1^2}{n\beta+\lambda}+\lambda\right)^{-1}. \tag{8.27}$$

We now want to show that if we have measured x^{mult} n times that the amount we can lower the conditional variance of $\mu(x^d)$ by observing x^{mult} again given in (8.24) is smaller than the amount given in (8.27) if we observe a new point x^{near}.

$$\begin{aligned}
&\left(\beta_2 - \frac{n\beta_0\beta_1}{n\beta+\lambda}\right)^2 \left(\beta - \frac{n\beta_1^2}{n\beta+\lambda} + \lambda\right)^{-1} \\
&= \left(\frac{\beta_2(n\beta+\lambda) - n\beta_0\beta_1}{n\beta+\lambda}\right)^2 \left(\frac{(\beta+\lambda)(n\beta+\lambda) - n\beta_1^2}{n\beta+\lambda}\right)^{-1} \\
&= \frac{\left(\beta_2(n\beta+\lambda) - n\beta_0\beta_1\right)^2}{(n\beta+\lambda)\left((\beta+\lambda)(n\beta+\lambda) - n\beta_1^2\right)} \\
&\geq \frac{\left(\beta_0(n\beta+\lambda) - n\beta_0\beta_1\right)^2}{(n\beta+\lambda)\left((\beta+\lambda)(n\beta+\lambda) - n\beta_1^2\right)} \qquad (8.28) \\
&\geq \frac{\left(\beta_0(n\beta+\lambda) - n\beta_0\beta\right)^2}{(n\beta+\lambda)\left((\beta+\lambda)(n\beta+\lambda) - n\beta^2\right)} \qquad (8.29) \\
&= \frac{\beta_0^2\lambda^2}{(n\beta+\lambda)(n\beta\lambda+\beta\lambda+\lambda^2)} \\
&= \frac{\beta_0^2\lambda}{(n\beta+\lambda)((n+1)\beta+\lambda)} \qquad (8.30)
\end{aligned}$$

In (8.28) we replaced β_2 with the smaller β_0. This is valid because the overall term is positive and the numerator is nonnegative because $\beta_0 \leq \beta_2$ and $\beta_1 \leq \beta$. In (8.29) we replaced β_1 with the larger β. This is valid because the derivative of (8.28) with respect to β_1 is negative. Using the quotient rule the derivative of (8.28) with respect to β_1 becomes:

$$\begin{aligned}
&\frac{(n\beta+\lambda)\left((\beta+\lambda)(n\beta+\lambda) - n\beta_1^2\right) 2(\beta_0(n\beta+\lambda) - n\beta_0\beta_1)(-n\beta_0) - (\beta_0(n\beta+\lambda) - n\beta_0\beta_1)^2 (n\beta+\lambda)(-2n\beta_1)}{c^2} \\
&= 2n(n\beta+\lambda)c^{-2}\left(\left((\beta+\lambda)(n\beta+\lambda) - n\beta_1^2\right)(\beta_0(n\beta+\lambda) - n\beta_0\beta_1)(-\beta_0) - (\beta_0(n\beta+\lambda) - n\beta_0\beta_1)^2(-\beta_1)\right) \\
&= 2n(n\beta+\lambda)c^{-2}\left((\beta_0(n\beta+\lambda) - n\beta_0\beta_1)^2\beta_1 - \left((\beta+\lambda)(n\beta+\lambda) - n\beta_1^2\right)(\beta_0(n\beta+\lambda) - n\beta_0\beta_1)\beta_0\right) \\
&= 2n(n\beta+\lambda)c^{-2}\left((n\beta+\lambda-n\beta_1)^2\beta_0^2\beta_1 - \left((\beta+\lambda)(n\beta+\lambda) - n\beta_1^2\right)(n\beta+\lambda-n\beta_1)\beta_0^2\right) \\
&= 2n(n\beta+\lambda)c^{-2}\beta_0^2(n\beta+\lambda-n\beta_1)\left((n\beta+\lambda-n\beta_1)\beta_1 - \left((\beta+\lambda)(n\beta+\lambda) - n\beta_1^2\right)\right) \\
&= 2n(n\beta+\lambda)c^{-2}\beta_0^2(n\beta+\lambda-n\beta_1)\left((n\beta+\lambda)\beta_1 - (\beta+\lambda)(n\beta+\lambda)\right) \\
&= \underbrace{2n(n\beta+\lambda)^2c^{-2}\beta_0^2}_{\geq 0}\underbrace{(n\beta+\lambda-n\beta_1)}_{\geq 0}\underbrace{(\beta_1-\beta-\lambda)}_{\leq 0} \\
&\leq 0.
\end{aligned}$$

We have now shown that if we have measured x^{mult} n times that the amount we can lower the conditional variance of $\mu(x^d)$ by observing x^{mult} again given in (8.24) is smaller than the amount given in (8.27) if we observe a new point x^{near}. This is true for $n = 0, 1, 2, ...$ so using an induction argument we see $max_{x_0,\ldots,x_{n-1}\in B(x^{acc},\epsilon)} Var^n[\mu(x^d)]$ equals $Var^n[\mu(x^d)]$ under π^{mult}. □