THE KNOWLEDGE GRADIENT ALGORITHM USING
LOCALLY PARAMETRIC APPROXIMATIONS

Bolong Cheng

Arta A. Jamshidi
Warren B. Powell

Electrical Engineering
Princeton University
Princeton, NJ 08544, USA

Operations Research and Financial Engineering
Princeton University
Princeton, NJ 08544, USA

ABSTRACT

We are interested in maximizing a general (but continuous) function where observations are noisy and may be expensive. We derive a knowledge gradient policy, which chooses measurements which maximize the expected value of information, while using a locally parametric belief model which uses linear approximations around regions of the function, known as clouds. The method, called DC-RBF (Dirichlet Clouds with Radial Basis Functions) is well suited to recursive estimation, and uses a compact representation of the function which avoids storing the entire history. Our technique allows for correlated beliefs within adjacent subsets of the alternatives and does not pose any a priori assumption on the global shape of the underlying function. Experimental work suggests that the method adapts to a range of arbitrary, continuous functions, and appears to reliably find the optimal solution.

1   INTRODUCTION

We consider the problem of maximizing an unknown but continuous function over a finite set of possible alternatives (which may be sampled from a continuous region), where observations of the function are noisy and may be expensive to compute. This problem arises under settings such as simulation-optimization, stochastic search, and ranking and selection. A popular strategy involves response surface methods which fit polynomial approximations to guide the search of the next. Our work was motivated by difficulties we encountered fitting parametric surfaces, even to relatively simple functions. Low order approximations can produce terrible approximations, while higher order models quickly suffer from over fitting. This experience led us to consider a variety of statistical strategies, but ultimately produced a new local parametric procedure called Dirichlet Clouds with Radial Basis Functions (DC-RBF), developed by Jamshidi and Powell (2013). This paper addresses the problem of doing stochastic search using the knowledge gradient, where the underlying belief model is represented using DC-RBF.

The optimization of noisy functions, broadly referred to as stochastic search, has been studied thoroughly since the seminal paper Robbins and Monro (1951) which introduces the idea of derivative-based stochastic gradient algorithms. For an extensive coverage of the literature for stochastic search methods see, e.g., Spall (2003), Sutton and Barto (1998), and Fu (2006). A separate line of research has evolved under the umbrella of active (or optimal) learning where observations are made specifically based on some measure of value of information, see Powell and Ryzhov (2012) for a review. The idea of making measurements based on the marginal value of information is introduced by Gupta and Miescke (1996) and extended under the name knowledge gradient using a Bayesian approach which estimates the value of measuring an alternative by the predictive distributions of the means, as shown in Frazier et al. (2008). The online learning setting with discrete alternatives is studied in Gittins (1979); the knowledge gradient is extended to online (bandit) settings in Ryzhov et al. (2012). The case of continuous decisions has been studied in Agrawal (1995)

and Ginebra and Clayton (1995). Most of the previous work in ranking and selection problems assume the alternatives to be independent (alternatives close to each other do not exhibit correlated beliefs), see Nelson et al. (2001).

There is a small literature that deals with correlated beliefs (Frazier et al. 2009; Huang et al. 2006). Villemonteix et al. (2009) introduces entropy minimization-based methods for Gaussian processes. An adaptation of the knowledge with correlated beliefs using kernel regression and aggregation of kernels for estimating the belief function is proposed in Barut and Powell (2013). This work builds on the work presented in Mes et al. (2011), where the estimates are the hierarchical aggregates of the values. This work is an extension of the knowledge gradient with linear beliefs, given in Negoescu et al. (2011), to non-parametric beliefs.

There are three major classes of function approximation methods: look-up tables, parametric models (linear or non-linear), and nonparametric models. Parametric regression techniques such as linear regression (see Montgomery et al. (2001)) assume that the underlying structure of the data is known a priori and is in the span of the regressor function. Due to the simplicity of this approach it is commonly used for regression.

Nonparametric models, such as Eubank (1988), Müller (1988), offer the attraction of considerable generality by using the raw data to build local approximations of the function, producing a flexible but data-intensive representation. Nonparametric models are less sensitive to structural errors arising from a parametric model. Most nonparametric models require keeping track of all observed data points, which make function evaluations increasingly expensive as the algorithm progresses, a serious problem in stochastic search. Bayesian techniques for function approximation or regression are computationally intensive and require storage of all the data points (see Gelman et al. (2003)). Local polynomial models, such as Fan (1992), build linear models around each observation and keep track of all the data points.

Another class of approximation algorithms use local approximations around regions of the function, rather than each prior observation. Radial basis functions have attracted considerable attention due to their simplicity and generality. One of the main attractions of the radial basis functions (RBFs) is that the resulting optimization problem can be broken efficiently into linear and nonlinear subproblems. Normalized RBFs are presented in Jones et al. (1990) which perform well for approximation for smaller amount of training data. For a comprehensive treatment on various growing RBF techniques and automatic function approximation technique using RBF, see Jamshidi and Kirby (2007) and references therein.

DC-RBF is motivated by the need to approximate functions within stochastic search algorithms where new observations arrive recursively. As we obtain new information from each iteration, DC-RBF provides a fast and flexible method for updating the approximation. DC-RBF is more flexible than classical parametric models, and provides a compact representation to minimize computational overhead. Unlike similar algorithms in the literature, our method has only one tunable parameter, assuming that the input data has been properly scaled.

The main contribution of this paper is the derivation of the knowledge gradient using DC-RBF to create the belief model. We show the performance of this proposed idea in the context of several illustrative examples. Section 2 formulates the ranking and selection model and establishes the notation used in this paper. This section also highlights the Bayesian inference that is needed to build the approximation. Section 3 reviews the knowledge gradient using both lookup table and a linear, parametric belief model. Section 4 reviews the DC-RBF approximation technique that is used for constructing the belief model in this paper. Section 5 derives the knowledge gradient using the DC-RBF belie model. Section 6 demonstrates the performance of the proposed methodology using examples drawn from different problem classes. Finally Section 7 provides the concluding remarks and highlights our future work.

2   THE RANKING AND SELECTION MODEL

We consider a finite set of alternatives $\mathscr{X} = \{1, 2, ..., M\}$ where our belief about the true value μ_x of each alternative $x \in \mathscr{X}$ is multivariate normal, given by $\mu \sim \mathscr{N}(\theta^0, \Sigma^0)$, where (θ^0, Σ^0) represents the prior mean

and covariance matrix. If we choose to observe $x = x^n$, we observe $\hat{y}_x^{n+1} = \mu_x + \varepsilon^{n+1}$, where $\mathbb{E}^n[\mu] = \theta^n$ and where ε^{n+1} is a random variable with known variance λ_x or, equivalently, precision $\beta_x^\varepsilon = (\lambda_x)^{-1}$. We define μ to be the vector of all the means $[\mu_1, \ldots, \mu_M]'$. Having a fixed budget of N measurements, we need to make sequential sampling decisions $x^0, x^1, \ldots, x^{N-1}$ to learn about these alternatives.

As a result of this sequential sampling framework, it is natural to define the filtration $\mathscr{F}^n$ as the σ-algebra generated by $\{(x^0, \hat{y}_{x^0}^1), (x^1, \hat{y}_{x^1}^2), \ldots, (x^{n-1}, \hat{y}_{x^{n-1}}^n)\}$. Let $\mathbb{E}^n$ and Var^n denote $\mathbb{E}[\cdot | \mathscr{F}^n]$ and $Var[\cdot | \mathscr{F}^n]$ (the conditional expectation and variance with respect to $\mathscr{F}^n$), respectively.

In the off-line setting, the goal is to find the optimal alternative after N measurements, where the final decision is

$$x^N = \arg\max_{x \in \mathscr{X}} \theta_x^N.$$

Let Π be the set of all possible measurement policies, and $\mathbb{E}^\pi$ be the expectation taken when the policy $\pi \in \Pi$ is used. The problem to find the optimal policy can be written as

$$\sup_{\pi \in \Pi} \mathbb{E}^\pi \left[\max_{x \in X} \theta_x^N \right].$$

We use the Bayesian setting to sequentially update the estimate of the alternatives. If at time n we choose $x^n = x$, we observe $\hat{y}_x^{n+1}$. This action along with the corresponding observation, and our prior belief of μ, we can compute the $n+1$st posterior distribution using standard normal sampling with a multivariate normal prior distribution using the following updating equations, Gelman et al. (2003):

$$\theta^{n+1} = \theta^n + \frac{\hat{y}^{n+1} - \theta_x^n}{\lambda_x + \Sigma_{xx}^n} \Sigma^n e_x, \tag{1}$$

$$\Sigma^{n+1} = \Sigma^n - \frac{\Sigma^n e_x e_x' \Sigma^n}{\lambda_x + \Sigma_{xx}^n}, \tag{2}$$

where e_x is the standard basis vector. We can further rearrange Equation (1) as the time n conditional distribution of θ^{n+1},

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

where $\tilde{\sigma}(\Sigma^n, x^n) = \frac{\Sigma^n e_x}{\sqrt{\lambda_x + \Sigma_{xx}^n}}$ and Z is a standard normal random variable.

3 KNOWLEDGE GRADIENT WITH CORRELATED BELIEFS

In this work we focus on the knowledge gradient with correlated beliefs, which is a sequential decision policy for learning correlated alternatives, Frazier et al. (2009). In this framework the state of knowledge at time n is defined as $S^n := (\theta^n, \Sigma^n)$. The goal is to pick the best possible option if we stop measuring at time n. The value of being in state S^n is defined as

$$V^n(S^n) = \max_{x' \in \mathscr{X}} \theta_{x'}^n.$$

If we choose to measure $x^n = x$, a new belief state $S^{n+1}(x)$ is reached after observing $\hat{y}_x^{n+1}$ using the updating equations (1) and (2). The value of this new state is

$$V^{n+1}(S^{n+1}(x)) = \max_{x' \in \mathscr{X}} \theta_{x'}^{n+1}.$$

Note that the incremental value due to measuring alternative x is defined as:

$$\begin{aligned} v_x^{KG,n} &= \mathbb{E}\left[V^{n+1}(S^{n+1}(x)) - V^n(S^n) | S^n, x^n = x\right], \\ &= \mathbb{E}\left[\max_{x' \in \mathscr{X}} \theta_{x'}^{n+1} | S^n, x^n = x\right] - \max_{x' \in \mathscr{X}} \theta_{x'}^n. \end{aligned}$$

We would like to maximize the expected value of $V^{n+1}(S^{n+1}(x))$ at time n. The knowledge gradient policy chooses a measurement that would lead to maximize the expected incremental value, i.e.,

$$x^{KG,n} = \arg\max_{x \in \mathcal{X}} v_x^{KG,n}.$$

An algorithm to compute the KG values for alternatives with correlated beliefs is demonstrated in Frazier et al. (2009) which is briefly described here. Consider an alternative way to write the KG formula as

$$
\begin{aligned}
\mathcal{V}_x^{KG,n} &= h\left(\theta^n, \tilde{\sigma}(\Sigma^n, x)\right), \\
&= \mathbb{E}\left[\max_{x' \in \mathcal{X}} \theta_{x'}^n + \tilde{\sigma}_{x'}(\Sigma^n, x^n)Z \,|\, S^n, x^n = x\right] - \max_{x' \in \mathcal{X}} \theta_{x'}^n,
\end{aligned}
$$

where $h(a,b) = \mathbb{E}\left[\max_i a_i + b_i Z\right] - \max_i a_i$ is a generic function for any vector a and b of the same dimension. The expectation can be computed as the point-wise maximum of affine functions $a_i + b_i Z$ with an algorithm of complexity $O(M^2 \log(M))$.

The algorithm sorts the alternatives with b_i in increasing order, then removes terms (a_i, b_i) if there is some i' such that $b_i = b_{i'}$ and $a_i > a_{i'}$ (i.e., removing parallel slopes with lower constant intercepts). Then it removes dominated $(a_{i'}, b_{i'})$ if for all $Z \in \mathbb{R}$ there exists some i such that $i \neq i'$ and $a_{i'} + b_{i'}Z \leq a_i + b_i Z$. After all of the dominated components are dropped, new vectors $(\tilde{a}, \tilde{b})$ of dimension $\tilde{M}$ are created. At the end of this process, we are left with a concave set of affine functions.

The function $h(a,b)$ can be computed via

$$h(a,b) = \sum_{i=1,\dots,\tilde{M}} (\tilde{b}_{i+1} - \tilde{b}_i) f\left(-\left|\frac{\tilde{a}_i - \tilde{a}_{i+1}}{\tilde{b}_{i+1} - \tilde{b}_i}\right|\right), \tag{3}$$

where $f(z) = \phi(z) + z\Phi(z)$. Here, $\phi(z)$ and $\Phi(z)$ are the normal density and cumulative distribution functions respectively, Frazier et al. (2009).

Now suppose we can represent the true μ as a linear combination of a set of parameters, i.e., $\mu = X\alpha$, where elements of α are the coefficients of the column vectors of a design matrix X. Instead of maintaining a belief on the alternatives, we can maintain a belief on the coefficients, which is generally of a much lower dimension. If we have a multivariate normal distribution on $\alpha \sim \mathcal{N}(\vartheta, C)$, we can induce a normal distribution on μ, namely, $\mu \sim \mathcal{N}(X\vartheta, XCX^T)$, Negoescu et al. (2010). Note that we use the scripted ϑ for the estimate of coefficients to differentiate them from the alternatives described in the previous section. This linear transformation applies for prior and posterior distributions.

If at time n, alternative x is measured, let $\tilde{x}^n$ be the row vector of X corresponding to alternative x. We can update ϑ^{n+1} and C^{n+1} recursively via

$$
\begin{aligned}
\vartheta^{n+1} &= \vartheta^n + \frac{\hat{\varepsilon}^{n+1}}{\gamma^n} C^n \tilde{x}^n, \\
C^{n+1} &= C^n - \frac{1}{\gamma^n}\left(C^n \tilde{x}^n (\tilde{x}^n)^T C^n\right).
\end{aligned}
$$

This is useful if, for example, we are trying to find the best set of a vector of parameters which might have 10 or 20 dimensions. These equations allow us to generate tens of thousands of potential measurement alternatives x. If we use a lookup table belief model as was used in Frazier et al. (2009), this would involve creating and updating a matrix Σ^n with tens of thousands of rows and columns. Using a parametric belief model, we only need to maintain the covariance matrix C^n, which is determined by the dimensionality of the parameter vector ϑ. We never need to compute the full matrix XCX^T, although we will have to compute a row of this matrix.

4 DC-RBF AS BELIEF MODEL

The DC-RBF scheme introduces a cover over the input space to define local regions, as described in Jamshidi and Powell (2013). This scheme only stores a statistical representation of data in local regions and locally approximates the data with a low order polynomial such as a linear model. The local model parameters are quickly updated using recursive least squares. A nonlinear weighting system is associated with each local model which determines the contribution of this local model to the overall model output. The combined effect of the local approximations and the weights associated to them produces a nonlinear function approximation tool. On the data sets that we have studied, this technique is efficient, both in terms of computational time and memory requirements compared to classical nonparametric regression methods. The new method is robust in the presence of homoscedastic and heteroscedastic noise. Our method automatically determines the model order for a given dataset (one of the most challenging tasks in nonlinear function approximation). Unlike similar algorithms in the literature, DC-RBF method has only one tunable parameter; in addition it is asymptotically unbiased and provides an upper bound on the bias for finite samples Jamshidi and Powell (2013).

Instead of the linear relationship $\mu = X\alpha$, we assume there is a non-linear relationship between μ and a,

$$\mu = \frac{\sum_{i=1}^{N_c} \psi_i(c_i, W_i) X \alpha_i}{\sum_{i=1}^{N_c} \psi_i(c_i, W_i)}, \tag{4}$$

where ψ_i is a kernel function with parameters center c_i and width W_i.

In our model, we employ the Gaussian kernel function

$$\psi(c, W) = \exp(-||x - c||_W^2),$$

where $||x||_W = \sqrt{x^T W x}$. N_c is the total number of kernels, which is automatically determined based on the shape of the function.

At time n, if alternative $x^n = x$ is chosen, we observe $\hat{y}_x^{n+1}$. According to the DC-RBF algorithm, the data point x^n needs to be assigned to a cloud; the resulting cloud I is determined by the distance between the input data point x^n and the center of the current clouds,

$$D_i = ||x^n - c_i||.$$

Let $I = \arg\min_i D_I$. If $D_I > D_T$, (D_T is a given known threshold distance), we create a new cloud U_{N_c+1} (where N_c denotes the number of existing clouds), otherwise we will update cloud U_I with the new data point, Jamshidi and Powell (2013). We observe that this logic is independent of the observation $\hat{y}_x^{n+1}$ which simplifies (but also limits) our algorithm. However, the important simplification is that the set of clouds at time $n+1$, given the clouds at n, is a deterministic function of x^n, and this greatly simplifies our adaptation of the knowledge gradient.

To update a particular cloud U_I, we need to update the local regression model and its weight. Locally, the iteration counter for cloud I is k_I where $\sum_i k_i = n$, and $\tilde{x}^n$ is the attribute vector (note that this is a row vector from the data matrix X). The recursive update for the response follows from the linear regression update

$$\vartheta_I^{k_I+1} = \vartheta_I^{k_I} + \frac{\hat{\varepsilon}^{k_I+1}}{\gamma^{k_I}} C_I^{k_I} \tilde{x}^n,$$

$$C_I^{k_I+1} = C_I^{k_I} - \frac{1}{\gamma^{k_I}} \left(C_I^{k_I} \tilde{x}^n (\tilde{x}^n)^T C_I^{k_I} \right),$$

where we define $\hat{\varepsilon}^{n+1} = \hat{y}^{n+1} - (\vartheta_I^n)^T \tilde{x}^n$ and $\gamma^n = \lambda_{x^n} + (\tilde{x}^n)^T C_I^n \tilde{x}^n$.

The center of the weighting kernel is updated via the following equation

$$c_I^{k_I+1} = c_I^{k_I} + \frac{x^n - c_I^{k_I}}{k_I}.$$

Then the width of the weighting kernel is updated via the Welford formula, given in Knuth (1997). Note that W_I is computed for each dimension separately below

$$
\begin{aligned}
Q^{k_I} &= Q^{k_I-1} + (x^n - Q^{k_I})/k_I, \\
S^{k_I} &= S^{k_I-1} + (x^n - Q^{k_I-1})(x^k - Q^{k_I}).
\end{aligned}
$$

The k_I-th estimate of the variance is $W_I^2 = \frac{1}{k_I-1} S^{k_I}$.

5 KNOWLEDGE GRADIENT WITH DC-RBF BELIEF MODEL

It is apparent that Equation (4) is a weighted sum of multiple linear estimates. From the above updating equations of DC-RBF, each data point is used to update the local estimate of one cloud, implying the coefficient vectors $\alpha_i, i = 1, \cdots, N_c$ are independent. Suppose we impose normal beliefs on each α_i such that $\alpha_i \sim \mathcal{N}(\vartheta_i, C_i)$; we note that μ is the linear combination of independent normal random variables, so it is also a normal random variable, where its distribution is expressed as

$$\mu \sim \mathcal{N}\left(\sum_{i=1}^{N_c} T_i \vartheta_i, \sum_{i=1}^{N_c} T_i C_i T_i^T\right),$$

where $T_i = \frac{\psi_i X}{\sum_{i=1}^{N_c} \psi_i}$.

Substituting this new expression of the belief state back to $h(a,b) = \mathbb{E}\left[\max_i a_i + b_i Z\right] - \max_i a_i$, where

$$
\begin{aligned}
a &= \sum_{i=1}^{N_c} T_i^n \vartheta_i^n, \\
b &= \frac{\sum_{i=1}^{N_c} T_i^n C_i^n T_i^{n,T} e_x}{\sqrt{\lambda_x + \sum_{i=1}^{N_c} (T_i^n C_i^n T_i^{n,T})_{xx}}}.
\end{aligned}
$$

We can compute the KG factor again via the correlated belief method described in Equation (3).

6 EMPIRICAL RESULTS

Here we demonstrate the performance of the algorithm on a variety of synthesized data sets. The data sets have distinct features in terms of complexity of the response function as well as the noise. Note that the algorithm provides a functional representation of the underlying belief function.

6.1 Quadratic Functions

Here we demonstrate KG-DC-RBF maximizing over a quadratic function on the interval $[1, 10]$. We discretize the interval by 0.1 producing a domain with 91 alternatives. We set the sampling variance as $\lambda = 1$ and the distance threshold for DC-RBF, $D_T = 2$. We initialize with a constant prior, with $\theta^0 = \max_x(\mu_x) + 3\lambda$ and $\Sigma^0 = 100I$, where I is the identity matrix. Figure 1 illustrates the estimate of the function produced by KG-DC-RBF after 50 iterations. We observe that the proposed algorithm has performed well in finding the maximum of the given function. There are 5 components in the DC-RBF model.
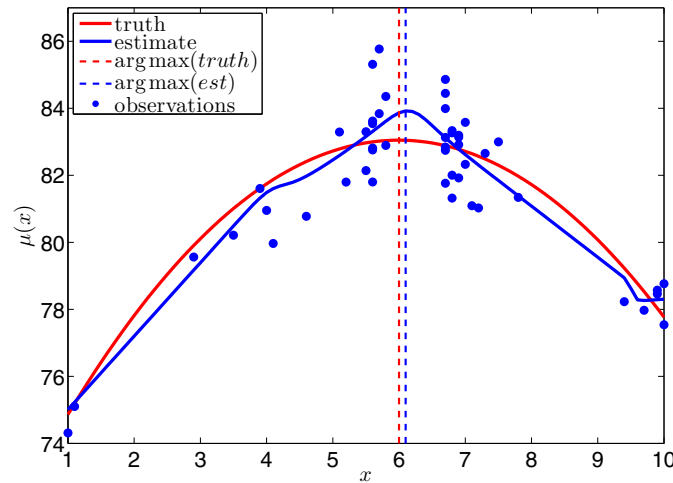
Figure 1: Estimate of a quadratic function with KG-DC-RBF, after 50 iterations. This estimate aims at finding the maximum of the underlying function from a set of noisy observations and does not aim at approximating the original function as a whole.

6.2 Newsvendor Problems

In the newsvendor problem, a newsvendor is trying to determine how many units of an item x to stock. The stocking cost and selling price for the item is c and p, respectively. One could assume an arbitrary distribution on demand D. The expected profit is given by

$$f(x) = \mathbb{E}\left[p\min(x,D)\right] - cx.$$

With known prices and an uncertain demand, we want the optimal stocking quantity that maximizes profit.

This problem represents a nontrivial challenge for many online estimation methods. First, it is highly heteroscedastic, with much higher measurement noise for larger values of x (especially $x \geq \mathbb{E}D$). Second, if $p - c$ is small relative to c, the function is highly nonsymmetric, which causes problems for simple regressors such as quadratic functions. Higher order approximations suffer from overfitting and generally lose the essential concave property of the function.

Figure 2 shows an instance of this problem, where we use $p = 4$, $c = 3.9$, and D is sampled from $\mathcal{N}(40, 1)$. The small variance of the distribution causes the function to have a sharp drop, whereas a larger variance yields a smoother decline. In this particular example, we set the measurement noise $\lambda = 1$, along with prior beliefs $\theta^0 = \max_x(\mu_x) + 3\lambda$ and $\Sigma^0 = 100I$. In this example, we see that KG-DC-RBF quickly converges to the optimal alternative. This is otherwise very difficult if we have a parametric belief model such as a quadratic function.

6.3 Performance on One-dimensional Test Functions

In this section we compare the performance KG-DC-RBF with two other methods of the same class. The first is Pure Exploration, where each alternative is selected with probability $1/M$ at every time step. The second is knowledge gradient with non-parametric beliefs (KG-NP) presented in **Barut and Powell (2013)**, which uses aggregates over a set of kernel functions to estimate the truth. We tested these policies on two types of Gaussian Processes. All GP functions are defined on mesh points $x = 1, 2, \ldots, 100$. For all the policies, we select the prior mean to be $\theta^0 = \max_i \mu_i + 3\lambda$, and prior covariance to be $\Sigma^0 = 100I$. For KG-NP, we use the Epanechnikov kernel and pick the number of the kernel as 64.
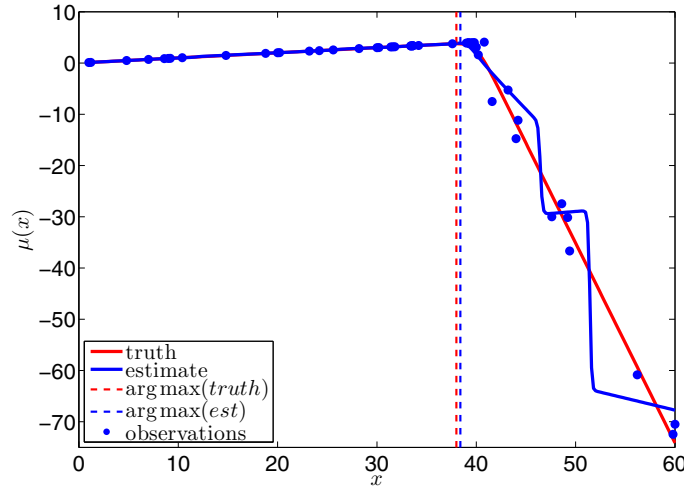
Figure 2: Estimate of the optimal order quantity for a newsvender problem using KG-DC-RBF with $D_T = 5$.

6.4 Gaussian Process with Homoscedastic Covariance Functions

In this part we test KG-DC-RBF policies on Gaussian processes with the covariance function

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

which produces a stationary process with variance σ^2 and length scale ρ. Higher values of ρ produce fewer numbers of peaks, resulting in a smoother function. The variance σ^2 scales the function vertically. Figure 3 illustrates randomly generated Gaussian processes with different values of ρ.
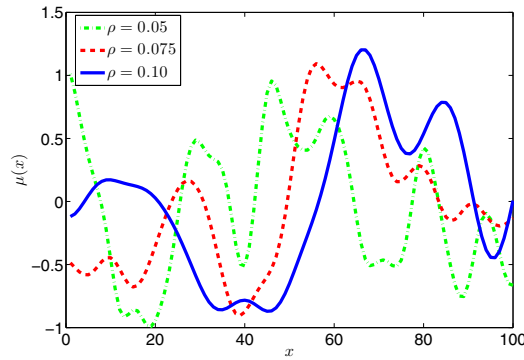

Figure 3: Stationary Gaussian processes with different values of ρ.

For our test functions, we use $\sigma^2 = 0.5$ and choose $\rho \in [0.05, 0.1]$ and $D_T = 5$. For both of these values, we generate 500 test functions to test the policy. We use the opportunity cost as the performance benchmark,

$$OC(n) = \max_i \mu_i - \theta_{i^\star},$$

with $i^\star = \arg\max_x \theta_x^n$. This measures the difference between the maximum of the truth and the value of the best alternative found by a given algorithm. We take the average of the opportunity cost for each different parameters over ρ. We also test the policies on two different sampling noise levels. The lower one with

$\lambda = 0.01$ and the high noise with $\lambda = \frac{1}{4}(\max(\mu) - \min(\mu))$. The opportunity costs on a log scale for different policies are given in Figure 4.

(a) $\rho = 0.1, \lambda = 0.01$

(b) $\rho = 0.1, \lambda = d/4$

(c) $\rho = 0.05, \lambda = 0.01$
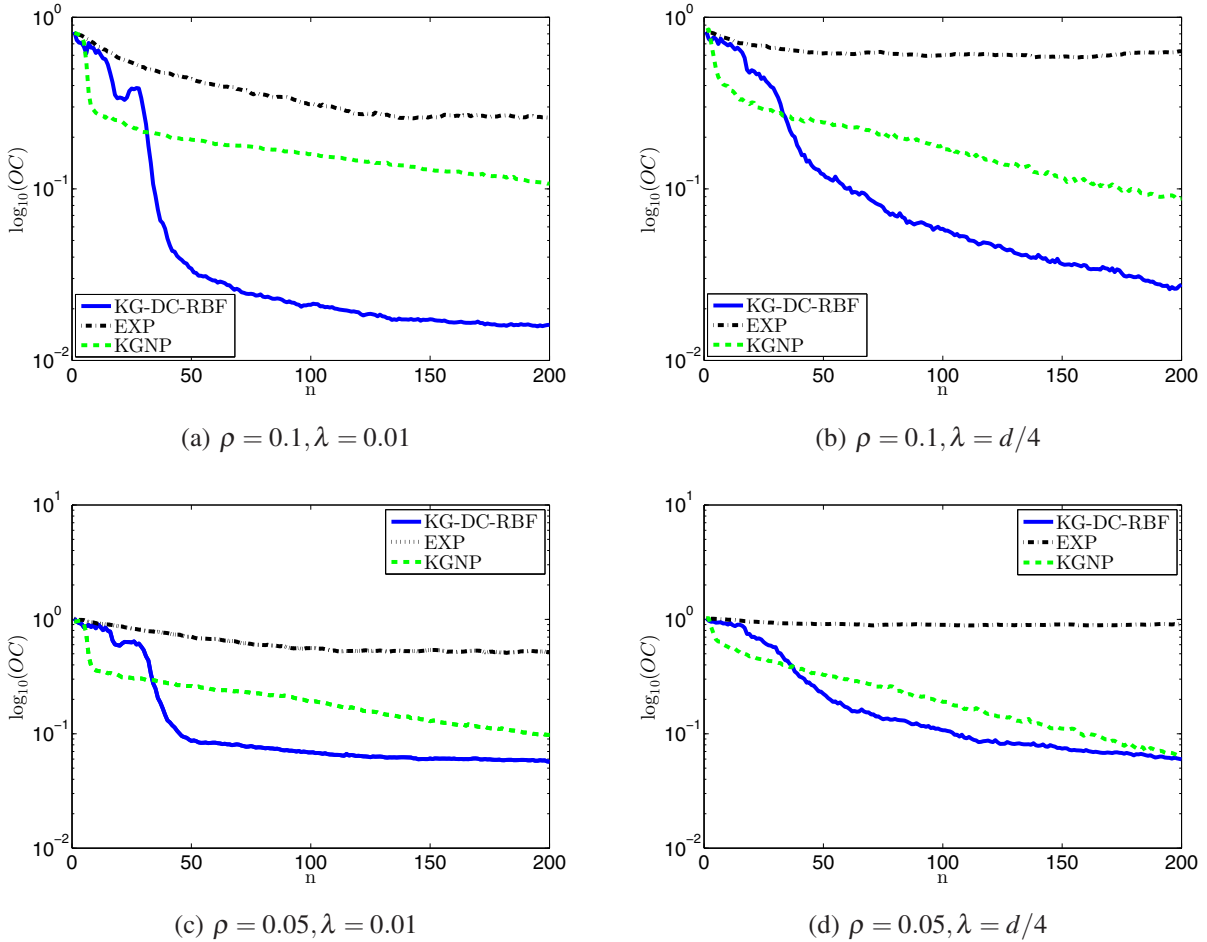
(d) $\rho = 0.05, \lambda = d/4$

Figure 4: Comparison of policies on homoscedastic GP, where $d = \max(\mu) - \min(\mu)$.

The KG-DC-RBF method outperforms both KG-NP and EXP in the low noise setting ($\lambda = 0.01$), due the fact that it could construct the underlying belief model with a few observations. In the high noise setting, with $\rho = 0.05$, the functions have more peaks and valleys, and the advantage of KG-DC-RBF becomes less apparent with the fixed $D_T = 5$, while keeping a slightly better performance. The high sampling variance makes some of the smaller peaks and valleys indistinguishable and requires more alternatives to sample from, making it difficult to identify local maxima.

6.4.1 Gaussian Process with Heteroscedastic Covariance Functions

In this part we consider non-stationary covariance functions, particularly the Gibbs covariance function, Gibbs (1997). It has a similar structure to the exponential covariance function, as described above, but is heteroscedastic. The Gibbs covariance function is given by

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

where $l(i)$ is an arbitrary positive function in i. Here we choose a horizontally shifted periodic sine function given by

$$l(i) = 10\left(1 + \sin\left(2\pi\left(\frac{i}{100} + u\right)\right)\right) + 1,$$

where u is a random number from $[0,1]$. The opportunity costs on a log scale for different policies are given in Figure 5.

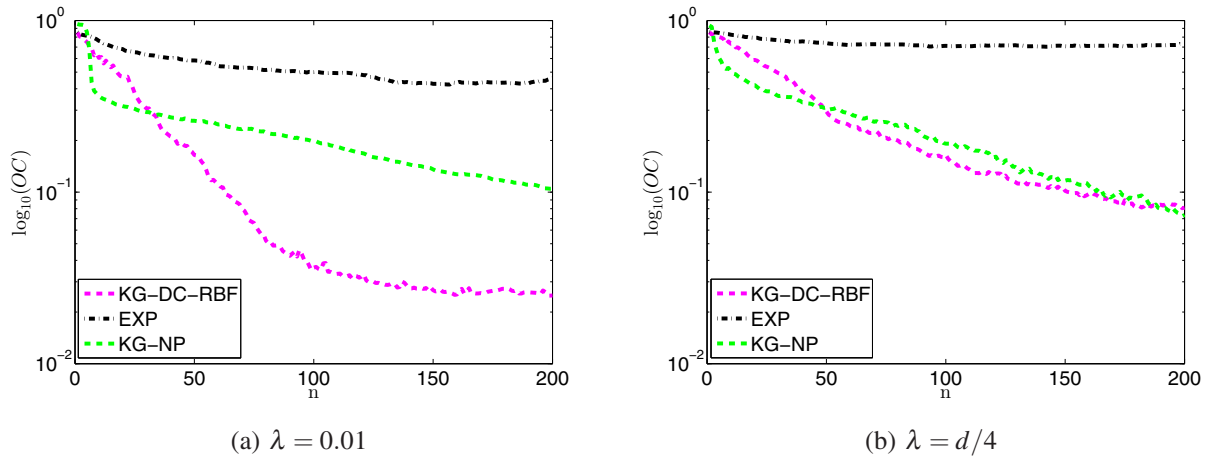(a) $\lambda = 0.01$ (b) $\lambda = d/4$

Figure 5: Comparison of policies on heteroscedastic GP, where $d = \max(\mu) - \min(\mu)$.

Similar to the previous experiment in the stationary case, we observe that the KG-DC-RBF outperforms both KG-NP and EXP in the low noise setting. Note that in the heteroscedastic case the covariance between alternatives changes according to the index of the alternative, which makes it a harder estimation problem. Similar to the previous experiment we observe that in the high noise case, the advantage of KG-DC-RBF becomes less apparent with the fixed $D_T = 2$, while keeping a slightly better performance. These experiments show that the KG-DC-RBF method is robust in the heteroscedastic case. In addition, the KG-DC-RBF is faster and requires less storage.

7 CONCLUSION

In this paper we introduced the adaptation of the knowledge gradient technique for optimizing functions using the DC-RBF belief model. The knowledge gradient offers an efficient search strategy by maximizing the marginal value of a measurement. We note that the method focuses on finding the optimal solution rather than producing a good approximation of the function (even near the optimum).

The method offers some useful features.

- It overcomes the need for extensive tuning as required by parametric approximations. The local parametric approach automatically adapts to the order of the function.
- The belief model naturally accommodates heteroscedastic noise.
- We avoid the need to store the entire history of observations, but we do require storing history in the form of a smaller set of clouds.
- We also avoid the need to specify hyperparameters for Bayesian priors.
- The updating of the belief model is fully recursive, and the knowledge gradient is quite easy to compute using the knowledge gradient for correlated beliefs using the linear belief model for each cloud.

- The proposed technique can be used both for on-line and off-line learning problems by using the simple bridging formula given in Ryzhov et al. (2012).

The experimental work shown here is encouraging. Not surprisingly, we get better performance on functions with relatively low noise in the measurement process. Our experiments were all performed on relatively low-dimensional functions, and it remains to be seen how it would scale to higher dimensional problems. In addition, we have found that the fitting of linear models to clouds with a small number of datapoints can be highly unreliable, especially in the presence of high levels of measurement noise. For this reason, we are investigating the use of a hierarchical blend of constant and linear functions with weights that adapt to the quality of the fit.

REFERENCES

Agrawal, R. 1995. "The Continuum-Armed Bandit Problem". *SIAM J. Control Optimization* 33:1926–1951.

Barut, E., and W. B. Powell. 2013. "Optimal Learning for Sequential Sampling with Non-Parametric Beliefs". *J. of Global Optimization*:(to appear)

Eubank, R. L. 1988. *Spline Smoothing and Nonparametric Regression*. Marcel Dekker, New York.

Fan, J. 1992, Dec. "Design-Adaptive Nonparametric Regression". *J. of the American Statistical Association* 87 (420): 998–1004.

Frazier, P., W. Powell, and S. Dayanik. 2009, May. "The Knowledge-Gradient Policy for Correlated Normal Beliefs". *INFORMS J. on Computing* 21 (4): 599–613.

Frazier, P. I., W. B. Powell, and S. Dayanik. 2008. "A Knowledge-Gradient Policy for Sequential Information Collection". *SIAM J. Control Optim.* 47 (5): 2410–2439.

Fu, M. C. 2006. *Handbooks in Operations Research and Management Science*, Chapter 19 Gradient Estimation, in Simulation, 575–616. Elsevier.

Gelman, A., J. B. Carlin, H. S. Stern, and D. B. Rubin. 2003. *Bayesian Data Analysis*. second ed. Chapman and Hall/CRC.

Gibbs, M. 1997. *Bayesian Gaussian Processes for Regression and Classification*. Ph. D. thesis, University of Cambridge.

Ginebra, J., and M. K. Clayton. 1995. "Response Surface Bandits". *J. of the Royal Statistical Society, Series B (Methodological)* 57:771–784.

Gittins, J. C. 1979. "Bandit Processes and Dynamic Allocation Indices". *J. of the Royal Statistical Society, Series B (Methodological)* 41:148-177.

Gupta, S. S., and K. J. Miescke. 1996. "Bayesian Look Ahead One-Stage Sampling Allocations for Selection of the Best Population". *J. of Statistical Planning and Inference* 54:229–244.

Huang, D., T. T. Allen, W. I. Notz, and N. Zeng. 2006. "Global Optimization of Stochastic Black-Box Systems Via Sequential Kriging Meta-Models". *J. of Global Optimization* 34:441–466.

Jamshidi, A. A., and M. J. Kirby. 2007. "Towards a Black Box Algorithm for Nonlinear Function Approximation over High-Dimensional Domains". *SIAM J. of Scientific Computation* 29 (3): 941–963.

Jamshidi, A. A., and W. B. Powell. "A Recursive Local Polynomial Approximation Method using Dirichlet Clouds and Radial Basis Functions". *SIAM J. on Scientific Computation*:submitted

Jones, R. D., Y. Lee, C. W. Barnes, G. W. Flake, K. Lee, P. S. Lewis, and S. Qian. 1990. "Function Approximation and Time Series Prediction with Neural Networks". *IJCNN International Joint Conference on Neural Networks* 1:649–665.

Knuth, D. E. 1997. *Art of Computer Programming*. 3rd ed, Volume 2. Addison-Wesley Professional.

Mes, M. R., W. B. Powell, and P. I. Frazier. 2011. "Hierarchical Knowledge Gradient for Sequential Sampling Hierarchical Knowledge Gradient for Sequential Sampling". *J. of Machine Learning Research* 12:2931-2974.

Montgomery, D. C., E. A. Peck, and G. G. Vining. 2001. *Introduction to Linear Regression Analysis*. 3 edition ed. Wiley-Interscience.

Müller, H. G. 1988. "Weighted Local Regression and Kernel Methods for Nonparametric Curve Fitting". *J. of the American Statistical Association* 82:231–238.

Negoescu, D. M., P. I. Frazier, and W. B. Powell. 2010. "The Knowledge-Gradient Algorithm for Sequencing Experiments in Drug Discovery". *INFORMS J. on Computing* 23 (3): 346–363.

Negoescu, D. M., P. I. Frazier, and W. B. Powell. 2011. "The Knowledge-Gradient Algorithm for Sequencing Experiments in Drug Discovery". *INFORMS J. on Computing* 23:346-363.

Nelson, B. L., J. Swann, D. Goldsman, and W. Song. 2001. "Simple Procedures for Selecting the Best Simulated System when the Number of Alternatives is Large". *Operations Research* 49:950-963.

Powell, W. B., and I. O. Ryzhov. 2012. *Optimal Learning*. John Wiley & Sons Inc.

Ryzhov, I. O., W. B. Powell, and P. I. Frazier. 2012. "The Knowledge Gradient for a General Class of Online Learning Problems". *Operations Research* 60:180–195.

Robbins, H., and S. Monro. 1951. "A Stochastic Approximation Method". *The Annals of Mathematical Statistics* 22:400–407.

Spall, J. C. 2003. *Introduction to Stochastic Search and Optimization*. John Wiley & Sons Inc.

Sutton, R. S., and A. G. Barto. 1998. *Introduction to Reinforcement Learning*. MIT Press.

Villemonteix, J., E. Vazquez, and E. Walter. 2009. "An Informational Approach to the Global Optimization of Expensive-to-Evaluate Functions". *J. of Global Optimization* 44:509-534.

AUTHOR BIOGRAPHIES

BOLONG (HARVEY) CHENG is a Ph.D. student in Electrical Engineering at Princeton University. He has a B.Sc. degree in Electrical Engineering and is doing research in stochastic optimization in energy. His e-mail is bcheng@princeton.edu.

ARTA A. JAMSHIDI is a Research Associate in the Department of Operations Research and Financial Engineering at Princeton University. He is an expert in function approximation. He earned his Ph.D. in Mathematics in 2008 developing novel nonlinear signal processing tools that serve various applications. His research interests include machine learning, signal and image processing, optimization and mathematical modeling. His email address is arta@princeton.edu.

WARREN B. POWELL is a Professor in the Department of Operations Research and Financial Engineering at Princeton University, and director of CASTLE Labs (http://www.castlelab.princeton.edu) which focuses on computational stochastic optimization and learning. He has taught at Princeton University since 1981. His email address is powell@princeton.edu and website is www.castlelab.princeton.edu.