Stratified Bayesian Optimization

Abstract

We consider derivative-free black-box global optimization of expensive noisy functions, when most of the randomness in the objective is produced by a few influential scalar random inputs. We present a new Bayesian global optimization algorithm, called Stratified Bayesian Optimization (SBO), which uses this strong dependence to improve performance. Our algorithm is similar in spirit to stratification, a classical technique from simulation, which uses strong dependence on a categorical representation of the random input to reduce variance. We demonstrate in numerical experiments that SBO outperforms a Bayesian optimization benchmark that does not take advantage of this dependence.

1. Introduction

We consider derivative-free black-box global optimization of expensive noisy functions,

$$\max_{x\in A\subset\mathbb{R}^n} \mathbb{E}\left[f\left(x,w,z\right)\right], \tag{1}$$

where the expectation is taken over $z \in \mathbb{R}^{d_1}$ and $w \in \mathbb{R}^{d_2}$, which have some joint probability density p, A is a simple compact set (e.g., a hyperrectangle, or a simplex), and we can directly observe only $f(x,w,z)$ at some collection of chosen or sampled x,w,z, and not its expectation, or the derivative of this expectation. We suppose that f has no special structural properties, e.g., concavity, or linearity, that we can exploit to solve this problem, making it a "black blox." We also suppose that evaluating f is costly or time-consuming, making these evaluations "expensive", severely limiting the number of evaluations we may perform. This typically occurs because each evaluation requires running a complex PDE-based or discrete-event simulation, or requires training a machine learning algorithm on a large dataset. When f comes from a discrete-event simulation, this problem is also often called "simulation optimization."

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

Bayesian optimization is a popular class of techniques for solving this problem, originating with the seminal paper (Kushner, 1964), and enjoying early contributions from (Mockus et al., 1978; Mockus, 1989). This class of techniques was popularized in the 1990s by the introduction in (Jones et al., 1998) of the most well-known Bayesian optimization method, Efficient Global Optimization (EGO), relying on earlier ideas from (Mockus, 1989). Recently the machine learning community has devoted considerable interest and effort to Bayesian optimization for its applications to tuning computationally intensive machine learning models, as in, e.g., (Snoek et al., 2012). Textbooks and surveys on Bayesian optimization include (Forrester et al., 2008; Brochu et al., 2010).

Most work on Bayesian optimization assumes we can observe the objective function directly without noise, but a substantial number of papers, e.g. (Villemonteix et al., 2009; Huang et al., 2006; Scott et al., 2011; Brochu et al., 2010), do allow noise and thus consider the problem (1). These methods all build a statistical model (usually using Gaussian processes) of the function $x \mapsto G(x) := \mathbb{E}[f(x,w,z)]$ using noisy observations, and then use an acquisition criterion, typically expected improvement or probability of improvement (Brochu et al., 2010), to decide where to sample next.

Existing work from Bayesian optimization for solving (1) rely on noisy evaluations in which w and z are drawn iid from their governing joint probability distribution p, and then $f(x,w,z)$ is observed. However, in many applications, we have the ability to choose not just x, but w as well, simulating the remaining components z conditioning on these values. (The choice of which random inputs to include in w and which in z was arbitrary above, but will be assumed below to accommodate this distinction.) For example, in a queuing simulation, we can simulate the individual arrival times of customers z conditioning on the overall number of arrivals w. We explore this in more detail in the numerical experiments. This ability to simulate random inputs given the value of some of their values is widely used in stratified sampling to estimate expectations with better precision (Glasserman, 2003).

This ability suggests the rephrasing of problem (1) into the equivalent problem

$$\max_{x\in A\subset\mathbb{R}^n} \mathbb{E}\left[F\left(x,w\right)\right] \tag{2}$$

where $F(x,w) := \mathbb{E}[f(x,w,z) \mid w]$, and the problems are equivalent because $\mathbb{E}[F(x,w)] = \mathbb{E}[f(x,w,z)] = G(x)$.

This equivalent formulation suggests that standard approaches to Bayesian optimization are wasteful from a statistical point of view, as they do not use past observations w to learn $G(x) = \mathbb{E}[F(x,w)]$, treating w only as an unobservable source of noise. Instead, one can use Bayesian quadrature (O'Hagan, 1991), which builds a Gaussian process model of the function $F(x,w)$ using past observations of $x, w, f(x,w,z)$, and then uses the known relationship $G(x) = \int F(x,w)p(w)\,dw$ (where we assume $p(w) := \int p(w,z)\,dz$ is known in closed form) to imply a second Gaussian process model on $G(x)$.

Using this ability, we develop an algorithm in this paper, called stratified Bayesian optimization (SBO), which chooses not just the x at which to evaluate $f(x,w,z)$, but also the w. It chooses these using an acquisition function discussed below, which is based on a value-of-information (Howard, 1966) calculation, and has a one-step Bayes-optimality property. It then samples z from its conditional distribution given w, and uses the resulting observation within a Bayesian quadrature framework to update its Gaussian process posterior on both $(x,w) \mapsto F(x,w)$ and $x \mapsto \mathbb{E}[F(x,w)]$. By using more information, we make our statistical model more powerful, and allow our optimization framework to provide better answers with fewer samples.

This approach is similar in spirit to stratified sampling (Glasserman, 2003), where our goal is to estimate $G(x) = E[F(x,w)] = E[f(x,w,z)]$ (for a fixed x, as this literature does not consider variation in x) and we choose which values of w at which to sample rather than sampling them from their marginal distribution, and then compensate for this choice via a known relationship between $F(x,w)$ and $G(x)$ to obtain lower variance estimates.

To decide how to choose x and w, we use a decision-theoretic approach which models the utility that results from solutions to the optimization problem (2). We then find the pair of values (x,w) at which to sample that maximizes the expected utility of the final solution, under the assumption, made for tractability, that we may take only one additional sample. Thus, our SBO algorithm is optimal in a decision-theoretic sense, in a one-step setting.

This one-step decision-theoretic approach follows the development of acquisition functions designed for other settings. In more traditional Bayesian optimization problems, the well-known expected improvement acquisition function (Mockus, 1989; Jones et al., 1998) has this optimality property when observations are noise-free and the final solution must be taken from previously evaluated solutions (Frazier & Wang, 2015), and the knowledge-gradient method (Frazier et al., 2009; Scott et al., 2011) has this optimality property when the final solution is not restricted to be a previously evaluated solution, in both the noisy and noise-free setting.

Our current approach significantly generalizes (Xie et al., 2012), which developed a similar method, but did not allow for the inclusion of unmodeled random inputs z, instead requiring all inputs to be included and modeled statistically in w. This introduces a heavy computational and statistical burden when dealing with problems in which the combined dimension of w and z is large, such as queuing problems and many other problems from engineering, significantly limiting its applicability.

This paper is organized as follows: In §2, we present the statistical model used for the problem. In §3, we present the SBO algorithm. In §3, we briefly give the main ideas of the computation of the value of information and its derivative. In §5, we present two numerical experiments. In §6, we conclude.

2. Statistical Model

The SBO algorithm that we will develop relies on a Gaussian process (GP) model of the underlying function F, which then implies (because integration is a linear function) a Gaussian process model over G. This statistical approach mirrors a standard Bayesian quadrature approach, but we summarize it here both to define notation used later, and because its application to Bayesian optimization is not standard.

We first place a Gaussian process prior distribution over the function F:

$$F(\cdot,\cdot) \sim GP(\mu_0(\cdot,\cdot), \Sigma_0(\cdot,\cdot,\cdot,\cdot)),$$

where μ_0 is a real-valued function taking arguments (x,w), and Σ_0 is a positive semi-definite function taking arguments (x,w,x',w'). Common choices for μ_0 and Σ_0 from the Gaussian process regression literature (Rasmussen & Williams, 2006; Murphy, 2012), e.g., setting μ_0 to a constant and letting Σ_0 be the squared exponential or Màtern kernel, are appropriate here as well.

Our algorithm will take samples sequentially. At each time $n = 1,2,\ldots,N$, our algorithm will choose x_n and w_n based on previous observations. It will then take M samples of $f(x_n,w_n,z)$ and observe the average response. More precisely, it will sample $z_{n,m} \sim p(z \mid w_n)$ for $m = 1,\ldots,M$ and observe $y_n = \frac{1}{M}\sum_{m=1}^{M} f(x_n,w_n,z_{n,m})$. The choice of M is an algorithm parameter, and should be chosen large enough that the central limit theorem may be applied, so that we may reasonably model the (conditional) distribution of y_n as normal. We will then have,

$$y_n | x_n, w_n \sim N\left(F(x_n,w_n), \sigma^2(x_n,w_n)/M\right),$$

where $\sigma^2(w,z) := \mathrm{Var}(f(x,w,z) \mid w)$. We assume that this conditional variance is finite for all x and w. In updating the posterior, we also assume that we observe this value $\sigma^2(x_n, w_n)$, although in practice we estimate it using the empirical variance from our M samples.

Let $H_n = (y_{1:n}, w_{1:n}, x_{1:n})$ be the history observed by time n. Then, the posterior distribution on F at time n is

$$F(\cdot,\cdot) \mid H_n \sim GP(\mu_n(\cdot,\cdot), \Sigma_n(\cdot,\cdot,\cdot,\cdot)),$$

where μ_n and Σ_n can be computed using standard results from Gaussian process regression (Rasmussen & Williams, 2006). To support later analysis, expressions for μ_n and Σ_n are provided in the supplement.

We denote by $\mathbb{E}_n$, Cov_n, and Var_n the conditional expectation, conditional covariance, and conditional variance on F (and thus also on G, since G is specified by F) with respect to the Gaussian process posterior given H_n. By results from Bayesian quadrature (O'Hagan, 1991), which rely on the previously noted fact that $G(x) = \int F(x,w)p(w)\,dw$, we have that

$$\mathbb{E}_n[G(x)] = \int \mu_n(x,w)p(w)\,dw := a_n(x), \tag{3}$$

$$\mathrm{Cov}_n\left(G(x), G(x')\right) = \int\int \Sigma_n\left(x,w,x',w'\right) p(w)\, p\left(w'\right) dw dw.' \tag{4}$$

Ignoring some technical details, the first line is derived using interchange of integral and expectation, as in $\mathbb{E}_n[G(x)] = \mathbb{E}_n[\int F(x,w)p(w)\,dw] = \int \mathbb{E}_n[F(x,w)p(w)]\,dw = \int \mu_n(x,w)p(w)\,dw$. The second line is derived similarly, though with more effort, by writing the covariance as an expectation, and interchanging expectation and integration.

3. Stratified Bayesian Optimization (SBO) Algorithm

Our SBO algorithm will choose points to evaluate using a value of information analysis (Howard, 1966), which maximizes the expected gain in the quality of the final solution to (1) that results from a sample.

To support this value of information analysis, we first consider the expected solution quality resulting for a particular set of samples. After n samples, if we were to choose the solution to (1) with the best expected quality with respect to the Bayesian posterior distribution on G, we would choose

$$x_n^* \in \arg\max_x \mathbb{E}_n[G(x)] = \arg\max_x a_n(x).$$

This is the Bayes-optimal solution when we are risk neutral. This solution has expected value (again, with respect to the posterior),

$$\mu_n^* := \max_x \mathbb{E}_n[G(x)] = \max_x a_n(x).$$

The improvement in expected solution quality that results from a sample at (x,w) at time n is

$$V_n(x,w) = \mathbb{E}_n\left[\mu_{n+1}^* - \mu_n^* \mid x_{n+1} = x, w_{n+1} = w\right]. \tag{5}$$

We refer to this quantity as the *value of information*, and if we have one evaluation remaining, then choosing to sample at the point with the largest value of information is optimal from a Bayesian decision-theoretic point of view. If we have more than one evaluation remaining, then it is not necessarily Bayes-optimal, but we argue that it remains a reasonable heuristic.

Thus, our Stratified Bayesian Optimization (SBO) algorithm is defined by

$$(x_{n+1}, w_{n+1}) \in \arg\max_{x,w} V_n(x,w). \tag{6}$$

Detailed computation of this value of information, and its gradient with respect to x and w, is discussed below in §4. We use this gradient to solve (6) using multi-start gradient ascent.

The SBO algorithm is summarized here:

Algorithm 1 SBO Algorithm

1: **First stage of samples** Evaluate F at n_0 points, chosen uniformly at random from A. Use maximum likelihood or maximum a posteriori estimation to fit the parameters of the GP prior on F, conditioned on these n_0 samples. Let μ_0 and Σ_0 be the mean function and covariance kernel of the resulting GP posterior on F.
2: **Main stage of samples:**
3: **for** $n = 1$ **to** N **do**
4: Update our Gaussian process posterior on F using all samples from the first stage, and samples $x_{1:n}$,$w_{1:n}$, $y_{1:n}$. This allows computation of μ_n and Σ_n as described in the supplement, computation of a_n through (3), and computation of V_n and ∇V_n as described in Section 4.
5: Solve $(x_{n+1}, w_{n+1}) \in \arg\max_{x,w} V_n(x,w)$ using multi-start gradient ascent and the ability to compute ∇V_n. Let (x_{n+1}, w_{n+1}) be the resulting maximizer.
6: Evaluate $y_{n+1} = \frac{1}{M}\sum_{m=1}^{M} f(x_{n+1}, w_{n+1}, z_{n+1,m})$ where $z_{n+1,m}$ are iid draws from $p(z \mid w_{n+1})$, and $p(z \mid w) = p(w,z)/p(w)$ is the conditional density of z given w.
7: **end for**
8: Return $x^* = \arg\max_x a_{N+1}(x)$.

4. Computation of the Value of Information and Its Gradient

In this section we discuss computation of the value information (5) and its gradient, to support implementation of the SBO algorithm.

We provide a table with the notation used in this section

Table 1: Table of Notation.

$G(x)$	$\triangleq$	$\mathbb{E}[f(x,w,z)]$
V_n	$\triangleq$	Value of Information at time n
$a_n(x)$	$\triangleq$	$\mathbb{E}_n[G(x)]$
H_n	$\triangleq$	History observed by time n
$B(x,i)$	$\triangleq$	$\int \Sigma_0(x,w,x_i,w_i)\,p(w)dw$, for $i=1,\ldots,n+1$
γ	$\triangleq$	$\begin{bmatrix} \Sigma_0(x_{n+1},w_{n+1},x_1,w_1) \\ \vdots \\ \Sigma_0(x_{n+1},w_{n+1},x_n,w_n) \end{bmatrix}$
A_n	$\triangleq$	$\left(\Sigma_0(x_i,w_i,x_j,w_j)\right)_{i,j=1}^n + \mathrm{diag}\left(\left(\sigma^2(x_i,w_i)\right)_{i=1}^n\right)$

4.1. Computation of the Value of Information

We first rewrite the value of information (5) as

$$V_n(x_{n+1},w_{n+1}) = \mathbb{E}_n\left[\max_{x'\in A} a_{n+1}(x') \mid x_{n+1},w_{n+1}\right] - \max_{x'\in A} a_n(x'). \quad (7)$$

To calculate this expectation, we must find the joint distribution of $a_{n+1}(x)$ across all x conditioned on (x_{n+1},w_{n+1}) and H_n for any x. This is provided by the following lemma.

Lemma 1 *There exists a standard normal random variable Z_{n+1} such that, for all x,*

$$a_{n+1}(x) = a_n(x) + \tilde{\sigma}_n(x,x_{n+1},w_{n+1})Z_{n+1}.$$

where

$$\tilde{\sigma}_n^2(x,x_{n+1},w_{n+1}) := Var_n[G(x)] - \mathbb{E}_n[Var_{n+1}[G(x)] \mid x_{n+1},w_{n+1}].$$

To compute the value of information, we then discretize the feasible set A, over which we take the maximum in (7), into $L<\infty$ points. We let A' denote this discrete set of points, so $A'\subseteq A$ and $|A'|=L$. For example, if A is a hyperrectangle, then we may discretize it using a uniform mesh.

Then, we approximate (7) by

$$\begin{aligned} V_n(x_{n+1},w_{n+1}) &= \mathbb{E}_n[\max_{x\in A} a_n(x) + \tilde{\sigma}(x,x_{n+1},w_{n+1})Z_{n+1}] - \max_{x\in A} a_n(x) \\ &\approx \mathbb{E}_n[\max_{x\in A'} a_n(x) + \tilde{\sigma}(x,x_{n+1},w_{n+1})Z_{n+1}] - \max_{x\in A'} a_n(x) \\ &= h(a_n(A'),\tilde{\sigma}_n(A',x_{n+1},w_{n+1})), \end{aligned}$$

where $a_n(A') = (a_n(x_i))_{i=1}^L$, $\tilde{\sigma}_n(x,w) = (\tilde{\sigma}_n(x_i,x,w))_{i=1}^L$, and $h:\mathbb{R}^L\times\mathbb{R}^L\to\mathbb{R}$ is defined by $h(a,b) = \mathbb{E}[\max_i a_i + b_i Z] - \max_i a_i$, where a and b are any deterministic vectors, and Z is a one-dimensional standard normal random variable. By convenience, we will denote $a_n(x_i)$ by e_i and $\tilde{\sigma}_n(x_i,x,w)$ by f_i for each i in $\{1,\ldots,L\}$. If $A=A'$, which is possible if A is a finite set, then the approximation in the second line above is exact.

In (Frazier et al., 2009), it is also shown how to compute h. Using the Algorithm 1 in that paper, we can get a subset of indexes $\{j_1,\ldots,j_\ell\}$ from $\{1,\ldots,L\}$, such that

$$\begin{aligned} V_n(x_{n+1},w_{n+1}) &= h(a_n(A'),\tilde{\sigma}_n(A',x_{n+1},w_{n+1})) \\ &= \sum_{i=1}^{\ell-1}\left(f_{j_{i+1}} - f_{j_i}\right) f(-|c_i|) \end{aligned}$$

where

$$\begin{aligned} f(z) &:= \varphi(z) + z\Phi(z), \\ c_i &:= \frac{e_{j_{i+1}} - e_{j_i}}{f_{j_{i+1}} - f_{j_i}},\ i=1,\ldots,\ell-1, \end{aligned}$$

and φ,Φ are the standard normal cdf and pdf, respectively. This shows how to compute the Value of Information V_n.

4.2. Computation of the Gradient of the Value of Information

We show how to compute the gradient of the Value of Information V_n in this section. Observe that if $\ell=1$, $V_n(x,w)=0$ and so $\nabla V_n\left(x,w^{(1)}\right)=0$. On the other hand, if $\ell>1$, we can see that

$$\nabla V_n(x,w) = \sum_{i=1}^{\ell-1}\left(-\nabla f_{j_{i+1}} + \nabla f_{j_i}\right)\varphi(|c_i|).$$

Consequently, we only need to compute ∇f_{j_i} for each i in $\{1,\ldots,\ell\}$. This can be done by observing that

$$\nabla\tilde{\sigma}_n(x,x_{n+1},w_{n+1}) = \beta_1\beta_3 - \frac{1}{2}\beta_1^3\beta_2[\beta_5-\beta_4]$$

where

$$\begin{aligned}
\beta_1 &= \left[\Sigma_0\left(x_{n+1}, w_{n+1}, x_{n+1}, w_{n+1}\right) - \gamma^T A_n^{-1}\gamma\right]^{-1/2}, \\
\beta_2 &= B\left(x, n+1\right) - \left[B\left(x,1\right) \ \cdots \ B\left(x,n\right)\right] A_n^{-1}\gamma, \\
\beta_3 &= \left(\nabla B\left(x, n+1\right) - \nabla\left(\gamma^T\right) A_n^{-1} \begin{bmatrix} B\left(x,1\right) \\ \vdots \\ B\left(x,n\right) \end{bmatrix}\right), \\
\beta_4 &= 2\nabla\left(\gamma^T\right) A_n^{-1}\gamma, \\
\beta_5 &= \nabla\Sigma_0\left(x_{n+1}, w_{n+1}, x_{n+1}, w_{n+1}\right).
\end{aligned}$$

4.3. Formulas for $\tilde{\sigma}_n\left(x, x_{n+1}, w_{n+1}\right)$ and $a_n\left(x\right)$

Here, we give expressions for a_n to compute the parameters of the posterior distribution of a_{n+1}. First, a_n can be computed using the following formula,

$$\begin{aligned}
a_n\left(x\right) &= \mathbb{E}\left[\mu_n\left(x,w\right)\right] \\
&= \mathbb{E}\left[\mu_0\left(x,w\right)\right] \\
&\quad + \left[B\left(x,1\right) \ \cdots \ B\left(x,n\right)\right] A_n^{-1} \begin{pmatrix} y_1 - \mu_0\left(x_1, w_1\right) \\ \vdots \\ y_n - \mu_0\left(x_n, w_n\right) \end{pmatrix}.
\end{aligned}$$

In some cases it is possible to get a closed-form formula for B, e.g. if w follows a normal distribution, the components of w are independent and we use the squared exponential kernel.

Finally, it is straightforward to see that the formula for $\tilde{\sigma}_n^2\left(x, x_{n+1}, w_{n+1}\right)$ is

$$\left[\frac{\left(B\left(x,n+1\right) - \left[B\left(x,1\right) \ \cdots \ B\left(x,n\right)\right] A_n^{-1}\gamma\right)}{\sqrt{\left(\Sigma_0\left(x_{n+1}, w_{n+1}, x_{n+1}, w_{n+1}\right) - \gamma^T A_n^{-1}\gamma\right)}}\right]^2.$$

5. Numerical Experiments

We now present simulation experiments illustrating how the SBO algorithm can be applied in practice, and comparing its performance against some baseline Bayesian optimization algorithms. We compare on a test problem with a simple analytic form (Section 5.1), on a realistic problem arising in the design of the New York City's Citi Bike system (Section 5.2), and also on simulated problems that show when SBO will be much better than one baseline Bayesian optimization algorithm (Section 5.3).

We consider two baseline Bayesian optimization algorithms. We use the Knowledge-Gradient policy of (Frazier et al., 2009) and Expected Improvement criterion (Jones et al., 1998), which both place the Gaussian process prior directly on $G(x)$, and use a standard sampling procedure, in which w and z are drawn from their joint distribution, and $f(x,w,z)$ is observed. Knowledge-Gradient policy is equivalent to SBO if all components of w are moved into z. Thus, comparing against KG quantifies the benefit of SBO's core contribution, while holding constant standard aspects of the Bayesian optimization approach.

We also solved the problems from (Section 5.1) and (Section 5.2) with Probability of Improvement (PI) (Brochu et al., 2010), but we did not include its results because both KG and EI outperformed PI. Moreover, EI's acquisition function is more satisfying than PI's acquisition function (Brochu et al., 2010).

When implementing the SBO algorithm, we use the squared exponential kernel,

$$\Sigma_0\left(x, w, x', w'\right) = \sigma_0^2 \exp\left(-\sum_{k=1}^{n} \alpha_1^{(k)}\left[x_k - x_k'\right]^2 - \sum_{k=1}^{d_1} \alpha_2^{(k)}\left[w_k - w_k'\left(1\right)\right]^2\right)$$

where σ_0^2 is the common prior variance, and $\alpha_1^{(1)}, \ldots, \alpha_1^{(n)}, \alpha_2^{(1)}, \ldots, \alpha_2^{(d_1)} \in \mathbb{R}_+$ are the length scales. These values, σ^2 and the mean μ_0 are calculated using maximum likelihood estimation following the first stage of samples of F.

5.1. Analytic Test Problem

In our first example, we consider the problem

$$\max_{x\in[-3,3]} \mathbb{E}\left[f\left(x,w,z\right)\right] = \max_{x\in[-3,3]} \mathbb{E}\left[-\frac{z}{w}x^2 - w\right]$$

where $w \sim N\left(0,1\right)$ and $z \mid w \sim N\left(w,1\right)$. It is easy to see that $F\left(x,w\right) = -x^2 - w$ and $G(x) = -x^2$.

Figure 2 illustrates and compares the progress of SBO and benchmark KG algorithm on this problem, and shows that modeling F, as does SBO, rather than modeling G directly, as does KG, provides a higher quality estimate of G.

Figure 3a compares the performance of SBO, KG and EI, plotting the number of samples beyond the first stage on the x axis, and the average true quality of the solutions provided, $G(\text{argmax}_x \mathbb{E}_n[G(x)])$, averaging over 3000 independent runs of the three algorithms.

5.2. New York City Citi Bike System

In our second example considers a queuing simulation based on New York City's Citi Bike system, in which system users may remove an available bike from a station at one location within the city, and ride it to a station with an available dock in some other location within the city. The optimization problem that we consider is the allocation of a constrained number of bikes (6000) to available

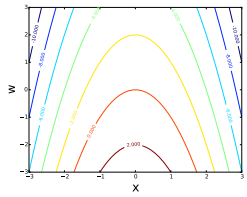
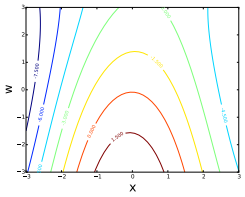
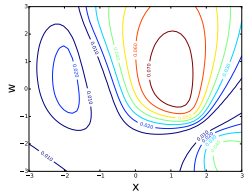


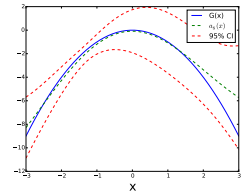
Figure 1: The contours of $F(x,w) = -x^2 - w$.



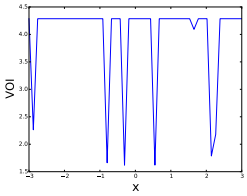
(a) SBO: The contours of SBO's estimate $\mu_n(x,w)$ of $F(x,w)$, at $n = 4$.



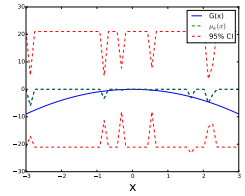
(b) SBO: The contours of the value of information $V_n(x,w)$ under SBO at $n = 4$. SBO's value of information depends on both x and w.



(c) SBO: The objective $G(x)$, and SBO's estimate $a_n(x)$ and 95% credible interval, at $n = 4$.



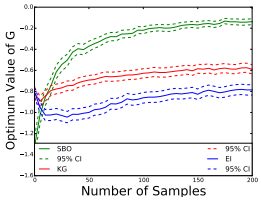
(d) KG: The value of information under KG. KG's value of information depends only on x.



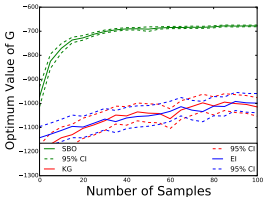
(e) KG: The objective $G(x)$, and KG's estimate $a_n(x)$ and 95% credible interval, at $n = 4$.

Figure 2: (**First row**) Left shows the contours of $F(x,w) = -x^2 - w$. Right shows the contours of SBO's estimate of F. SBO builds a statistical model of $F(x,w)$, which implies a statistical model on $G(x)$, rather than building a model on $G(x)$ directly like other Bayesian optimization methods. (**Second row**) Left shows the contours of SBO's value of information, which depends on both x and w, which SBO uses to choose the pair (x,w) to sample next. Right shows SBO's estimates of $G(x)$, which is based on the estimate of $F(x,w)$ in the first row. (**Third row**) Left shows KG's value of information, which depends only on x, which KG uses to choose the point x to sample next. Right shows KG's estimates of $G(x)$. This estimate is of lower quality than SBO's estimate above, because it does not use the observed values of w.

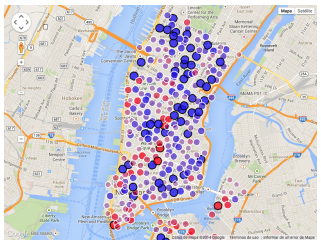
docks within the city at the start of rush hour, so as to minimize, in simulation, the expected number of potential trips in which the rider could not find an available bike at their preferred origination station, or could not find an available dock at their preferred destination station. We call such trips "negatively affected trips."



(a) Performance comparison between SBO and two Bayesian optimization benchmark, the KG and EI methods, on the analytic test problem from Section 5.1



(b) Performance comparison between SBO and two Bayesian optimization benchmark, the KG and EI methods, on the Citi Bike Problem from Section 5.2



(c) Location of bike stations (circles) in New York City, where size and color represent the ratio of available bikes to available docks.

Figure 3: Performance results for SBO on the analytic test problem (plot a), the Citi Bike problem (plot b), and a screenshot from our simulation of the Citi Bike problem (plot c).

We simulated in Python the demand of bike trips of a New York City's Bike System on any day from January 1st to December 30th between 7:00am and 11:00am. We used 329 actual bike stations, locations, and numbers of docks from the Citi Bike system, and estimated demand and average time for trips for every day in a year using publicly available data of the year 2014 from Citi Bike's website (cit).

We simulate the demand for trips between each pair of bike stations on a day using an independent Poisson process, and trip times between pairs of stations follows an exponential distribution. If a potential trip's origination station has no available bikes, then that trip does not occur, and we increment our count of negatively affected trips. If a trip does occur, and its preferred destination station does not have an available dock, then we also increment our count of negatively affected trips, and the bike is returned to the closest bike station with available docks.

We divided the bike stations in 4 groups using k-nearest neighbors, and let x be the number of bikes in each group

at 7:00 AM. We suppose that bikes are allocated uniformly among stations within a single group. Then we consider a directed graph between the bike stations, where each pair of bike stations has two directed edges, and we divided these edges in 4 groups. If the edge (i, j) is in a group, then (j, i) is also in that group. The random vector w is the number of bike trips in each of those groups during the period of our simulation. The random vector z contains all other random quantities within our simulation.

Figure 3b compares the performance of SBO, KG and EI, plotting the number of samples beyond the first stage on the x axis, and the average true quality of the solutions provided, $G(\text{argmax}_x \mathbb{E}_n[G(x)])$, averaging over 300 independent runs of the three algorithms. We see that SBO was able to quickly find an allocation of bikes to groups that attains a small expected number of negatively affected trips.

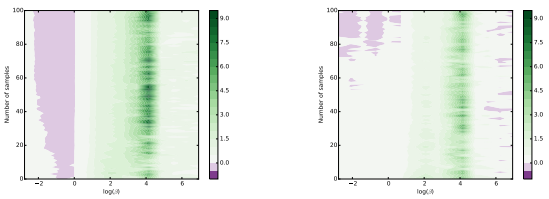
5.3. Simulated Problems

In our last examples, we demonstrate that SBO should be used when $\text{Var}[f(x,w,z)|f,x]/n$ is sufficiently larger than $\text{Var}[f(x,w,z)|f,x,w]/n$ where n is the number of samples taken per iteration, and $\text{Cov}[f(x,w,z), f(x,w',z)|f,x,z]$ is large enough. We define the difference between variances as $\text{Var}[f(x,w,z)|f,x]/n - \text{Var}[f(x,w,z)|f,x,w]/n$.

We consider the following simulated problems. We define functions $f(x,w,z) = h(x,w) + g(z)$ on $[0,1]^2 \times \mathbb{R}$ where $g(z)$ is drawn from a normal distribution with mean 0 and variance α_d, and h is drawn from a Gaussian Process with mean 0 and and covariance function $\Sigma((x,w),(x',w')) = \alpha_h \exp\left(-\beta \|(x,w)-(x',w')\|_2^2\right)$, where $\beta \in \left\{2^{-b} : b \in \{-4,-3,-2,-1,0,1,2,3,4\}\right\}$, $\alpha_h = \frac{A}{n}$ where $A \in \left\{\frac{1}{jn} : j \in \{2,4,8,16\}\right\}$ and $n \in \{1,2,4,8,16\}$ is the number of samples taken per iteration, and $\alpha_d = \frac{1}{n} - \alpha_h$. We also suppose that w is drawn uniformly from $\{0, 1/49, 2/49, \ldots, 1\}$. Consequently, we have that $\text{Var}[f(x,w,z)|f,x] = \alpha_d + \alpha_h = \frac{1}{n}$, $\text{Var}[f(x,w,z)|f,x,w] = \alpha_d$, and $\text{Cov}[f(x,w,z), f(x,w',z)|f,x,z] = \Sigma((x,w),(x,w'))$.

Figure 4a shows percentage increase between SBO and KG, when the difference between variances is 0.5. We see that SBO is always better than KG whenever $\beta >= 0.6$, which means that $\text{Cov}[f(x,w,z), f(x,w',z)|f,x,z]$ is large enough. However, KG is better than SBO when $\text{Cov}[f(x,w,z), f(x,w',z)|f,x,z]$ is small. This confirms the affirmation made at the beginning of the section.

Figure 4b shows percentage increase between SBO and KG, when the difference between variances is $1/1024$. We see that SBO KG are very similar. However, SBO improves when β increases, which means that $\text{Cov}[f(x,w,z), f(x,w',z)|f,x,z]$ increases. Since the difference between variances is very small, we see that there is no a big difference between the two algorithms, and KG is better than SBO when the total number of samples increases.



(a) Percentage increase between SBO and a Bayesian optimization benchmark, the KG method, when the difference between variances is 0.5

(b) Percentage increase between SBO and a Bayesian optimization benchmark, the KG method, when the difference between variances is $1/1024$

Figure 4: Performance results for SBO on simulated problems when the difference between variances is large enough (plot a), Performance results for SBO on simulated problems when the difference between variances is small (plot b).

By doing a factorial analysis, we see that β benefits SBO, A benefits KG, there is a positive interaction between β and A, n and A, and a negative interaction between β, n and A, whenever β is small enough.

6. Conclusion

We have presented a new algorithm called SBO for simulation optimization of noisy derivative-free expensive functions. This algorithm can be used with high dimensional random vectors, and it also outperforms the classical Bayesian approach to optimize functions in the examples presented.

Acknowledgements

Do not include acknowledgements in the initial version of the paper submitted for blind review.

References

Citi bike website. https://www.citibikenyc.com/, accessed May 2015.

Brochu, Eric, Cora, Vlad M, and De Freitas, Nando. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.

Forrester, Alexander, Sobester, Andras, and Keane, Andy.

Engineering design via surrogate modelling: a practical guide. John Wiley & Sons, 2008.

Frazier, Peter, Powell, Warren, and Dayanik, Savas. The knowledge-gradient policy for correlated normal beliefs. *INFORMS journal on Computing*, 21(4):599–613, 2009.

Frazier, Peter I. and Wang, Jialei. Bayesian optimization for materials design. *arXiv 1506.01349*, 2015.

Glasserman, Paul. *Monte Carlo methods in financial engineering*, volume 53. Springer Science & Business Media, 2003.

Howard, RA. Information Value Theory. *Systems Science and Cybernetics, IEEE Transactions on*, 2(1):22–26, 1966.

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

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

Kushner, H. J. A new method of locating the maximum of an arbitrary multi-peak curve in the presence of noise. *Journal of Basic Engineering*, 86:97–106, 1964.

Mockus, J. *Bayesian approach to global optimization: theory and applications*. Kluwer Academic, Dordrecht, 1989.

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

Murphy, Kevin P. *Machine learning: a probabilistic perspective*. MIT press, 2012.

O'Hagan, Anthony. Bayes–hermite quadrature. *Journal of statistical planning and inference*, 29(3):245–260, 1991.

Rasmussen, C.E. and Williams, C.K.I. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006. URL http://www.gaussianprocess.org/gpml.

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

Snoek, Jasper, Larochelle, Hugo, and Adams, Ryan P. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, pp. 2951–2959, 2012.

Villemonteix, Julien, Vazquez, Emmanuel, and Walter, Eric. An informational approach to the global optimization of expensive-to-evaluate functions. *Journal of Global Optimization*, 44(4):509–534, 2009.

Xie, J., Frazier, P.I., Sankaran, S., Marsden, A., and Elmohamed, S. Optimization of computationally expensive simulations with gaussian processes and parameter uncertainty: Application to cardiovascular surgery. In *50th Annual Allerton Conference on Communication, Control, and Computing*, 2012.