The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Bayesian Functional Optimization

Ngo Anh Vien
EEECS/ECIT,
Queen's University Belfast

Heiko Zimmermann
MLR, University of Stuttgart

Marc Toussaint
MLR, University of Stuttgart

Abstract

Bayesian optimization (BayesOpt) is a derivative-free approach for sequentially optimizing stochastic black-box functions. Standard BayesOpt, which has shown many successes in machine learning applications, assumes a finite dimensional domain which often is a parametric space. The parameter space is defined by the features used in the function approximations which are often selected manually. Therefore, the performance of BayesOpt inevitably depends on the quality of chosen features. This paper proposes a new Bayesian optimization framework that is able to optimize directly on the domain of function spaces. The resulting framework, *Bayesian Functional Optimization* (BFO), not only extends the application domains of BayesOpt to functional optimization problems but also relaxes the performance dependency on the chosen parameter space. We model the domain of functions as a reproducing kernel Hilbert space (RKHS), and use the notion of Gaussian processes on a real separable Hilbert space. As a result, we are able to define traditional improvement-based (PI and EI) and optimistic acquisition functions (UCB) as functionals. We propose to optimize the acquisition functionals using analytic functional gradients that are also proved to be functions in a RKHS. We evaluate BFO in three typical functional optimization tasks: i) a synthetic functional optimization problem, ii) optimizing activation functions for a multi-layer perceptron neural network, and iii) a reinforcement learning task whose policies are modeled in RKHS.

Introduction

Bayesian optimization is a derivative-free optimization scheme and is approached from the viewpoint of Bayesian theory (Jones, Schonlau, and Welch 1998; Brochu, Cora, and De Freitas 2010). It frames the optimization problem of unknown functions as a sequential decision task. These unknown functions are often costly to evaluate, especially in stochastic tasks, hence query points have to be selected such that the total cost of evaluations is optimized, for example, cost minimization in robotic control tasks as thoroughly discussed by (Deisenroth, Neumann, and Peters 2013), profit maximization in advertisement placement problems (Pandey, Chakrabarti, and Agarwal 2007), etc.. Specifically, the goal is to optimize an unknown real-valued objective function $f(x)$ on some bounded subset $\mathcal{X}$, mostly a subset of $\Re^d$. At each round t, the optimizer can only access a noisy evaluation y_t by querying $f(x_t)$ at a sample point x_t. The target is to find the minimizer x^* of the objective function while also minimizing the overall evaluation cost. To this end making decisions on an optimal next query point needs to integrate the information of all previous queries. A common approach of BayesOpt is to incorporate a Gaussian process prior as a probabilistic surrogate model of the unknown function. New candidate points are then sampled based on the posterior distribution of the learned model. Hence, the decision making process becomes a belief search problem. Many criteria can be used to exploit the distribution of this learned model, for example the probability of improvement (Kushner 1964), the expected improvement (Močkus 1975), the confidence bound criteria (Cox and John 1992; Srinivas et al. 2012), and information-based approaches (Hennig and Schuler 2012; Shahriari et al. 2014).

In this work, we are concerned with black-box optimization of *functional objectives*. By modeling candidate functions in reproducing kernel Hilbert space (RKHS) we enable BayesOpt to optimize in non-parametric, rich solution spaces while inheriting the useful structures and properties from a RKHS.

As a summary, our major contributions are three-fold: **First**, we propose the novel Bayesian functional optimization framework (**BFO**), that enables optimization in function spaces that are potentially infinite-dimensional. To this end BFO adopts a *functional Gaussian process prior* as a surrogate model for objective functionals, moreover we propose three acquisition functionals to select which function should be evaluated next: Probability of Improvement, Expected Improvement, and iGP-UCB functionals (infinite GP-UCB). **Second**, by assuming the domain of BFO is a RKHS $\mathcal{H}_K$ with a kernel K, we show that functional gradients of those acquisition functionals can be derived analytically. Moreover, those functional gradients are functions in $\mathcal{H}_K$ which results in an efficient functional gradient update. **Third**, we provide a cumulative regret bound for iGP-UCB.

There have recently been similar effort in proposing new machine learning frameworks for learning on functional data, such as functional regression by (Kadri et al. 2015),

modeling policies for reinforcement learning in RKHS by (Bagnell and Schneider 2003; Lever and Stafford 2015; Vien, Englert, and Toussaint 2016), representing motion trajectories in RKHS by (Marinho et al. 2016; Dong et al. 2016), or finding geodesic shortest paths for physical systems by (Kasim and Norreys 2016). Particularly, the last work by (Kasim and Norreys 2016) is the closest to our work in which they also tackle global functional optimization problems. This method extends the Simultaneous Optimistic Optimisation (SOO) approach proposed by (Munos 2011) to optimize functionals, which however have to resort to discretization. Another similar work by (Vien, Dang, and Chung 2017) has also tried to extend CMA-ES to functional optimization

Background

We first give a brief problem statement, then review background about Gaussian processes (GP) and BayesOpt with a GP prior.

In Bayesian optimization (Močkus 1975) (BayesOpt), we are interested in finding the maximum of a black-box function $f : \mathcal{D} \to \Re$ on some bounded domain $\mathcal{D}$, where $\mathcal{D} \subseteq \Re^n$. BayesOpt takes a probabilistic approach to model the objective function f whose uncertainty quantification can be exploited in making decisions on which query point x the objective $f(x)$ is evaluated next. Therefore, any BayesOpt method essentially consist of two major components. Firstly, a probabilistic surrogate model, often chosen to be a Gaussian Process, to represent the belief over the unknown objective $f(x)$, Secondly, an acquisition function which computes a utility value for candidate evaluation points from the posterior distribution in order to select the next optimal evaluation point.

Gaussian Process

Using Bayesian methods, one can infer a probabilistic model of f that can be queried for estimates of $f(x)$, it's confidence, and correlation with nearby regions. A Gaussian process (GP) prior is a generalized distribution over infinite many Gaussian random variables, of which each finite subset is distributed jointly normal (Rasmussen 2006). A GP is defined as $\mathcal{GP}(\mu(\cdot), k(\cdot,\cdot))$ with mean function μ and covariance kernel k that is presumably bounded: $k(x, x') \leq 1, \forall x \in D$. The kernel, i.e. covariance function, k encodes differentiability and smoothness properties of samples $f(x) \sim \mathcal{GP}(\mu(x), k(x,\cdot))$. Assuming that the noise $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$ is i.i.d. Gaussian, the posterior over f after t observations $\{x_i, y_i\}_{i=1}^t$ is Gaussian with mean, covariance, and variance as

$$\begin{aligned}
\mu_t(x) &= \mathbf{k}_t(x)^\top (\mathbf{K}_t + \sigma^2 \mathbf{I})^{-1} \mathbf{y}_t, \\
k_t(x, x') &= k(x, x') - \mathbf{k}_t^\top(x)(\mathbf{K}_t + \sigma^2 \mathbf{I})^{-1} \mathbf{k}_t(x'), \\
\sigma_t^2(x) &= k_t(x, x),
\end{aligned}$$

where $\mathbf{y}_t = [y_1, y_2, \cdots, y_t]^\top$ is the vector of previous observations, $\mathbf{k}_t(x) = [k(x, x_1), k(x, x_2), \cdots, k(x, x_t))]^\top$ the vector of pairwise kernels between current and previous query points, and $\mathbf{K}_t$ is the $t \times t$ Gram matrix of kernels $k(x_i, x_j), \forall i, j \in \{1, 2, \cdots, t\}$. For later analysis, a stationary kernel $k(x, x') = \hat{k}(|x - x'|)$ is used. Common examples of a stationary kernel are the squared exponential and Matérn kernels.

Bayesian Optimization: Acquisition Functions

We have discussed a probabilistic surrogate model used to represent the belief over the unknown function f. We now discuss the strategies to select a sequence of query points $x_{1:t}$ by defining an acquisition function $u : \mathcal{D} \to \Re$. Here we present and use three traditional acquisition functions (Brochu, Cora, and De Freitas 2010): probability of improvement (PI) (Kushner 1964), expected improvement (EI) (Lizotte 2008), and an upper confidence bound criteria (UCB) (Cox and John 1992; Auer, Cesa-Bianchi, and Fischer 2002). We denote $x_{\text{best}}, y_{\text{best}} = f(x_{\text{best}})$ the best evaluation until time t, and $\Phi(\cdot)$ the cumulative distribution function of the standard normal distribution denoted by $\phi(\cdot)$.

Probability of Improvement: This strategy selects the query point that maximizes the probability of improvement over the current best value, which is analytically computed as

$$u_{\text{PI}}(x) = \Pr(f(x) > y_{\text{best}}) = \Phi\left(\gamma(x)\right) \tag{1}$$

where $\gamma(x) = \frac{\mu_t(x) - y_{\text{best}}}{\sigma_t(x)}$.

Expected Improvement: This strategy selects the query point that maximizes the expected improvement over the current best. Similarly, this criteria has the analytic form

$$u_{\text{EI}}(x) = \sigma_t(x)\left(\gamma(x)\Phi(\gamma(x)) + \phi(\gamma(x))\right). \tag{2}$$

GP-UCB: this strategy selects the query point that maximizes the upper confidence bound criteria such that, when meeting some rather mild conditions, the cumulative regret is bounded, as introduced by (Srinivas et al. 2012).

$$u_{\text{UCB}}(x) = \mu_t(x) + \beta_t^{1/2}\sigma_t(x) \tag{3}$$

where $\beta_t = \nu\tau_t$ and ν is a hyperparameter. The selection of $\nu = 1$ and $\tau_t = 2\log(t^{n/2+2}\pi^2/3\delta)$ (n being the dimensionality of the domain $\mathcal{D}$) is shown to make GP-UCB achieve no-regret with probability $1 - \delta$.

There are a number of BayesOpt methods dealing with very high-dimensional problems, for example learning sparse additive model in high dimensions by (Tyagi, Gärtner, and Krause 2014; Tyagi et al. 2016), using random embeddings on high-dimensions by (Wang et al. 2016), learning a lower dimensional subspace by (Djolonga, Krause, and Cevher 2013). However, those methods work by assuming that the underlying problem is an inherently simple task (low-dimensional) but hidden in a very high dimensional space. On the other hand, none of those methods directly work with problems on function domains (potentially infinite-dimensional).

Functional Optimization: A Bayesian Approach

We now present BFO, a principled Bayesian optimization framework for both functional optimization tasks and optimization problems on non-parametric domains.

Algorithm 1 RKHS-REMBO

1: Generate a random bounded linear operator on RKHS, $T : \Re^d \to \mathcal{H}$
2: Define a bounded region set on $\mathcal{Z} \subset \Re^d$
3: Set $\mathcal{D}_0 = \emptyset$
4: **while** (not terminate) **do**
5: Select $z_{t+1} = \arg\max_{z \in \mathcal{Z}} u(z)$ (maximizing an acquisition function on $\mathcal{Z}$)
6: Sample $y_{t+1} = f(Tz_{t+1}) + \epsilon_{t+1}$
7: Update the data $\mathcal{D}_{t+1} = \mathcal{D}_t \cup \{z_{t+1}, y_{t+1}\}$
8: Tuning the kernel hyper-parameters of the $\mathcal{GP}$ on the domain $\mathcal{Z}$
9: **end while**

Problem Statement

We consider the problem of globally maximizing an unknown objective functional $f : \mathcal{H}_k \to \Re$, where $\mathcal{H}_k$ is a reproducing kernel Hilbert space with real-valued reproducing kernel $k : \mathcal{D} \times \mathcal{D} \to \Re$, $\mathcal{D} \subseteq \Re$, consisting of $span\{k(x, \cdot) \mid x \in \mathcal{D}\}$ and it's closure.

At each round t, a function $h_t \in \mathcal{H}$ is selected, and a noisy evaluation $y_t = f(h_t) + \epsilon_t$ is returned, where $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$.

Naive Approach

This section suggests one simple Bayesian optimization approach for the above problem, using the random embedding idea from (Wang et al. 2016) as described in Algorithm 1, called RKHS-REMBO (Bayesian Optimization on RKHS with Random Embedding). Many previous parametric Bayesian optimization approaches might be extended to tackle the above problem. However, we believe that those extensions using more sophisticated methods like learning a lower dimensional subspace by (Djolonga, Krause, and Cevher 2013), using the Karhunen-Loeve theorem (Jorgensen and Song 2007), or kernel learning (Wilson 2014)) might be non-trivial. Given a kernel function k, a random bounded linear operator T is generated by first sampling randomly d functions h_i in RKHS: $h_i = \sum_{i=1}^{N} \alpha_i k(x_i, \cdot)$, where $\alpha_i \in \Re, x_i \in \mathcal{D}$ are sampled randomly. The random bounded linear operator T is formed as $T = [h_1, h_2, \ldots, h_d]$. One can consider T as a $|\mathcal{H}| \times d$ matrix ($|\mathcal{H}|$ is the dimensionality of $\mathcal{H}$ which is potentially infinite). As T is constructed from a set of RKHS functions, we can easily conclude that T is a bounded operator.

Similar to the original REMBO algorithm, RKHS-REMBO assumes that the unknown function f has an intrinsically d-dimensional structure (where d must be treated as a hyperparameter), instead of $|\mathcal{H}|$ which might be potentially infinite. Therefore, RKHS-REMBO can only result in a sub-optimal solution function h^* that depends on a fixed set of initially randomly sampled functions h_i. However, though this projection may approximate the function domain crudely, it provides a simple and fast solution.

Bayesian Functional Optimization

Our proposed BFO framework is depicted in Algorithm 2. BFO is constructed based on two choices: i) a GP prior to track the belief over the unknown objective functional $f(h)$ and ii) an acquisition functional $h : \mathcal{H}_k \to \Re$.

Gaussian Process for Functional Domains There was little effort in using GP for functional data as in work of (Shi and Choi 2011) in which they define a GP kernel function on the inputs of parametric form. However, we assume specifically that the input space is a RKHS $\mathcal{H}_k$, which allows us to directly define a GP kernel over functions in $\mathcal{H}_k$.

We assume that a Gaussian process on a real separable Hilbert space $\mathcal{H}_k$ with a scalar reproducing kernel $k = \langle \cdot, \cdot \rangle_{\mathcal{H}}$ models a prior distribution on the unknown functional $f(h), h \in \mathcal{H}_k$. A stochastic process $f = \{f(h), h \in \mathcal{H}_k\}$ defined in a complete probability space $(\Omega, \mathcal{F}, \mu)$ is a Gaussian process if f is a Gaussian family of random variables such that $\mathrm{cov}(f(h), f(g)) = K(h, g)$, where $K(\cdot, \cdot)$ is required to be a positive semi-definite kernel $K : \mathcal{H}_k \times \mathcal{H}_k \to \Re$.

The kernel $K(h, g)$, where $h, g \in \mathcal{H}_k$, can be constructed based on RKHS kernels k evaluated at support points from the underlying RKHS function domain $\mathcal{D}$. We provide two simple functional GP kernels:

Polynomial kernel: $K(h, g) = (\langle h, g \rangle_{\mathcal{H}_k} + c_1)^{c_2}$ (where $c_1, c_2 \geq 0$ are hyper-parameters). Hence if h and g the form: $h = \sum_{i=1}^{N} \alpha_i k(x_i^{(h)}, \cdot)$, $g = \sum_{i=1}^{M} \beta_i k(x_i^{(g)}, \cdot)$, then the kernel $K(h, g)$ can be computed as

$$K(h, g) = \Big(\sum_{i=1,j=1}^{N,M} \alpha_i \beta_j k(x_i^{(h)}, x_j^{(g)}) + c_1 \Big)^{c_2}.$$

Stationary kernel: This kernel involves a computation of the distance $\|g - h\|^2 = \langle g - h, g - h \rangle_{\mathcal{H}_k}$, which again can be computed based on the evaluation of kernels $k(\cdot, \cdot)$ as

$$\begin{aligned} \langle g - h, g - h \rangle_{\mathcal{H}_k} = & \sum_{i=1,j=1}^{N,N} \alpha_i \alpha_j k(x_i^{(h)}, x_j^{(h)}) \\ & + \sum_{i=1,j=1}^{M,M} \beta_i \beta_j k(x_i^{(g)}, x_j^{(g)}) \\ & - 2 \sum_{i=1,j=1}^{N,M} \alpha_i \beta_j k(x_i^{(h)}, x_j^{(g)}) \end{aligned}$$

RBF kernel: An RBF kernel can easily be constricted using the stationary kernel as

$$K(g, h) = \exp(-\|g - h\|^2_{\mathcal{H}_k} / 2\sigma^2).$$

Posterior update: With a positive definite kernel K, the posterior over f is updated similarly to the standard GP. For a data set of noisy evaluations $\mathbf{y}_t = [y_1, y_2, \cdots, y_t]^\top$ and sampled functions $\{h_1, h_2, \cdots, h_t\}$, the posterior is again a GP distribution with mean functional $\mu_t(\cdot)$ and covariance kernel $K_t(\cdot, \cdot)$ as

$$\begin{aligned} \mu_t(h) &= \mathbf{k}_t(h)^\top (\mathbf{G}_t + \sigma^2 \mathbf{I})^{-1} \mathbf{y}_t \\ K_t(h, h') &= K(h, h') - \mathbf{k}_t^\top(h) (\mathbf{G}_t + \sigma^2 \mathbf{I})^{-1} \mathbf{k}_t(h') \\ \sigma_t^2(h) &= K_t(h, h) \end{aligned}$$

Algorithm 2 The BFO framework

1: Initialize $\mathcal{D}_0 = \emptyset$
2: Prior mean functional $\mu_0 \in \mathcal{H}_k$
3: **while** (not terminate) **do**
4: Select $h_{t+1} = \arg\max_{h \in \mathcal{H}_k} u(h)$ (maximizing the acquisition functional on $\mathcal{H}_k$)
5: Sparsify h_{t+1} to get a compact function $\tilde{h}_{t+1}$
6: Sample $y_{t+1} = f(\tilde{h}_{t+1}) + \epsilon_{t+1}$
7: Update the data $\mathcal{D}_{t+1} = \mathcal{D}_t \cup \{\tilde{h}_{t+1}, y_{t+1}\}$
8: Tuning the hyper-parameters of the kernel, $K : \mathcal{H}_k \times \mathcal{H}_k \to \Re$
9: **end while**

where $\mathbf{k}_t(h) = [K(h, h_1), K(h, h_2), h \cdots, K(h, h_t))]^\top$, and $\mathbf{G}_t$ is a $t \times t$ Gram matrix of functional kernels $K(h_i, h_j), \forall i, j \in \{1, 2, \cdots, t\}$.

Note that the above update is an extended Bayesian interpretation of the non-Bayesian kernel regression method on functional data as studied recently by (Kadri et al. 2015).

Acquisition Functionals We propose three acquisition functionals: probability of improvement, expected improvement, and iGP-UCB to select a function h to next evaluate $f(h)$. We denote $h_{\text{best}}, y_{\text{best}} = f(h_{\text{best}})$ the best evaluation until time t.

Probability of Improvement (PI):

$$u_{\text{PI}}(h) = \Phi(\gamma(h))$$

where the functional γ is defined as

$$\gamma(h) = \frac{\mu_t(h) - f(h_{\text{best}})}{\sigma_t(h)}$$

As $\Phi(\cdot)$ is a monotonically increasing function, maximizing PI can be replaced by just maximizing $\gamma(h)$.

Expected Improvement (EI):

$$u_{\text{EI}}(h) = \sigma_t(h)\big[\gamma(h)\Phi(\gamma(h)) + \phi(\gamma(h))\big]$$

iGP-UCB:

$$u_{\text{UCB}}(h) = \mu_t(h) + \beta_t^{1/2}\sigma_t(h)$$

Optimizing the above acquisition functionals might be hard. Fortunately, the functional gradient w.r.t functions on a RKHS $\mathcal{H}_k$ with a reproducing kernel k can be derived *analytically*

We are now computing the functional gradients of those acquisition functionals. Specifically, here we are stating the functional gradients for the iGP-UCB acquisition functional and RBF kernel $K(h, h') = \exp(-\|h' - h\|^2_{\mathcal{H}_k}/2\sigma^2)$ [1] We use the notion of the Fréchet derivative which is a derivative on Banach spaces. Let $\mathcal{V}$ and $\mathcal{W}$ be Banach spaces, and $U \in \mathcal{V}$ be an open subset of $\mathcal{V}$, then a function $f : U \to \mathcal{W}$ is called Fréchet differentiable at $h \in U$ if there exists a bounded linear operator $Df|_h : \mathcal{V} \to \mathcal{W}$ such that

$$\lim_{g \to 0} \frac{\|f(h+g) - f(h) - Df|_h(g)\|_{\mathcal{W}}}{\|g\|_{\mathcal{V}}} = 0$$

[1] We would like to refer the reader to the supplemenary material for the detailed analytic computation of the functional gradients of the three acquisition functionals.

Assumption 1 *Assume that each functional kernel $K(h_t, h)$ has a Fréchet derivative $Dh_t : \mathcal{H}_k \to \Re$*

According to (Chae 1985), when $\mathcal{W}$ is a real (or complex) space, the Fréchet derivative becomes a function in $\mathcal{H}_k$ i.e. $Df|_h \in \mathcal{H}_k$ and $Df|_h(g) = \langle Df|_h, g\rangle_{\mathcal{H}_k}$.

Lemma 1 *The Fréchet derivative at $h \in \mathcal{H}_k$ of the RBF kernel function $K(h_t, h) = \exp(-\|h_t - h\|^2_{\mathcal{H}_k}/2\sigma^2)$ is*

$$Dh_t|_h : g \mapsto \Big\langle \frac{K(h_t, h)}{\sigma^2}\big(h_t - h\big), g\Big\rangle_{\mathcal{H}_k}$$

As we can see the Fréchet derivative $Dh_t|_h$ at h of the RBF kernel is a function in $\mathcal{H}_k$ which support points are the combined set of suport points from h and h_t. Specifically, assuming that h and h_t have representation

$$h = \sum_{i=1}^{N_1} \alpha_i k(x_i, \cdot), \quad h_t = \sum_{i=1}^{N_2} \beta_i K(x'_i, \cdot)$$

then $Dh_t|_h$ is written as

$$Dh_t|_h(x) = \frac{K(h_t, h)}{\sigma^2}\sum_{i=1}^{N_2} \beta_i K(x'_i, x) - \frac{K(h_t, h)}{\sigma^2}\sum_{i=1}^{N_1} \alpha_i k(x_i, x)$$

where $x, x_i, x'_i \in \Re^n$.

Lemma 2 *The derivative of the mean and variance functionals at $h \in \mathcal{H}_k$ are the linear operators $D\mu_t|_h : \mathcal{H}_k \to \Re$, and $D\sigma_t^2|_h : \mathcal{H}_k \to \Re$ such that*

$$D\mu_t|_h(g) = D\mathbf{k}_t|_h(g)(\mathbf{G}_t + \sigma^2\mathbf{I})^{-1}\mathbf{y}_t$$
$$D\sigma_t^2|_h(g) = Dh|_h(g) - 2D\mathbf{k}_t|_h(g)^\top(\mathbf{G}_t + \sigma^2\mathbf{I})^{-1}\mathbf{k}_t(h)$$

where $D\mathbf{k}_t|_h$ is the Fréchet derivative of $\mathbf{k}_t(h)$, and $Dh|_h(g)$ is defined in Assumption 1. In addition, $D\mu_t|_h$ and $D\sigma_t^2|_h$ are functions in $\mathcal{H}_k$.

Proposition 1 *The Fréchet derivative of the UCB acquisition functional is*

$$Du_{UCB}|_h = D\mu_t|_h + \beta^{1/2}\frac{1}{\sigma_t^2(h)}D\sigma_t^2|_h$$

which is in $\mathcal{H}_k$.

Optimizing the acquisition functional: The recursive gradient update process starts with a randomly initialized function $h^{\{0\}} \in \mathcal{H}_K$. The function h is computed iteratively as

$$h^{\{l+1\}} = h^{\{l\}} + \alpha_l(Du|_{h^{\{l\}}} + \lambda h^{\{l\}}) \tag{4}$$

where α_l is a step-size, we denote $Du|_{h^{\{l\}}}$ the Fréchet derivative of the acquisition functionals u_{UCB}, u_{PI}, or u_{EI}. There are three key insight about this process.

1. The Lagrange function for acquisition functional optimization (using box constraints): we assume that there is a boundary on $\mathcal{H}_k$ that helps refrain global optimization methods from relentlessly exploring. Specifically, $\mathcal{H}_k$ consists of functions bounded by a constant C: $\|h\|_{\mathcal{H}_k} \leq C$. Therefore, we receive a functional optimization problem of a given acquisition function as

$$\max_{h \in \mathcal{H}_k} u(h) \quad s.t. \quad \|h\|_{\mathcal{H}_k} \leq C \tag{5}$$

We form the Lagrange function λ: $\max_{h \in \mathcal{H}_k} u(h) + \frac{\lambda}{2}(\|h\|^2_{\mathcal{H}_k} - C^2)$. Thus, we propose to optimize this function to find the next query function h, hence receive a functional gradient update as in Eq. 4. We treat λ as a hyperparameter.

2. Because all Fréchet derivatives of three acquisition functionals are in $\mathcal{H}_k$, the recursive update in Eq. 4 also results in functions $h^{\{l\}}$ in $\mathcal{H}_k$. Moreover, its representation depends on all support centres from $h_{1:t}$ and $h^{\{0\}}$. Specifically, assuming that

$$h_j = \sum_{i=1}^{N_l} \alpha_i^{\{j\}} k(x_i^{\{j\}}, \cdot) \quad \forall j \in (1, 2, \cdots, t),$$
$$h^{\{0\}} = \sum_{i=1}^{N} \alpha_i k(x_i, \cdot),$$

after l functional gradient updates in Eq. 4, $h^{\{l\}}$ might have representation as

$$h^{\{l\}} = \sum_{i=1}^{N} w_{0,i} \alpha_i k(x_i, \cdot) + \sum_{j=1}^{t} \sum_{i=1}^{N_j} w_{l-1,i} \alpha_i^{\{j\}} k(x_i^{\{j\}}, \cdot)$$

where w_{li} are the weights of the function $h^{\{l\}}$.

3. As realized by the above, the representation of the resulting optimal solution h^* depends on fixed support centres $x_i^{\{t\}}$ from previous sampled points h_t and centres x_i from the initial function $h^{\{0\}}$. Therefore, we propose to initialize $h^{\{0\}}$ randomly by a randomly sampled number of samples N (large enough), and random samples $x_{1:N}, \alpha_{1:N}$ to result in $h^{\{0\}} = \sum_{i=1}^{N} \alpha_i k(x_i, \cdot)$.

We use a multi-start strategy that reruns the above optimization process multiple times with randomly sampled functions $h^{\{0\}}$ to assure h^* is the global solution in optimizing the acquisition functional.

As the final solution $h_{t+1} = h^{\{*\}}$ might be a complex function (N_{t+1} large) (Step 4 in Algorithm 2), it might slow down the update of our GP in RKHS. We suggest to sparsify h_{t+1} before evaluating $f(h_{t+1})$. Our paper uses the kernel pursuit matching algorithm by (Vincent and Bengio 2002) to sparsify h to be represented by only d centres, instead of being $N + \sum_{j=1}^{t} N_j$. Sparsification is seen at Step 5 in Algorithm 2. After each iteration, we tune the hyperparameter (Step 8) of the kernel K (currently by maximizing the marginal likelihood).

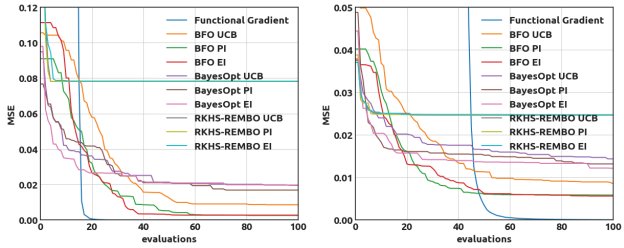


Figure 1: (left) MSE on 1D domain, (right) MSE on 2D domain

Theoretical Results The PI approach is known to be a heuristic rule as discussed by (Jones 2001), and the EI approach was recently proved to converge by (Vazquez and Bect 2010) and (Bull 2011) with limited assumptions about a fixed Gaussian process prior of finite smoothness and known smoothness of f, respectively. Therefore, we decide to provide only a theoretical result for the BFO UCB (iGP-UCB) method where the cumulative regret R_T is a performance metric. (Srinivas et al. 2012) provide cumulative regret bounds that depend on the dimensionality of the input space $\Re^n$. We follow their proof and provide a new regret bound suitable for our problem setting. The main difficulty is to deal with the (potentially) infinite dimensional domain. Many results of (Srinivas et al. 2012) can only hold with assuming finite dimensionality. First we define the maximum information gain γ_T after T rounds as

$$\gamma_T = \max_{H \subset \mathcal{H}_k, |H|=T} \mathrm{I}(\mathbf{y}_H; \mathbf{f}_H) = \frac{1}{2} \log |\mathbf{I} + \sigma^{-2} \mathbf{G}_T| \tag{6}$$

where $\mathbf{G}_T$ is the covariance matrix $K(h, h')$ for $h, h' \in H$. In the following theorem, we show the bound on the cumulative regret for iGP-UCB.

Theorem 1 *Define* $C_1 = 8/\log(1 + \sigma^{-2})$, *we have:* $R_T \leq \sqrt{C_1 T \beta_T \gamma_T} + \frac{\pi^2}{6}, \quad \forall T \geq 1$, *with probability greater than* $1 - \delta$, *where* $\beta_T = 2d_0 \log(r d_0 b T^2 \pi_t \sqrt{\log(2d_0 a)/\delta}) - 2d_0 \log(1 - 2\epsilon t^2)$ *in which* d_0, r, a, b *are parameters depending on discretization of the search space and* $\sum_{t \geq 1}^{T} 1/\pi_t = 1, \pi_t > 0$.

For a sketch of the proof, we use the Stone-Weierstrass theorem twice in order to approximate any function $f(h), \forall h \in \mathcal{H}_k$ by a parametric function, which has a finite-dimensional domain based on two stages. The first stage approximates f by a finite set of basis $\{h_i\}_{i=1}^{d}$. This step is equivalent to represent f in parametric form as $f(h) \approx \sum_{i=1}^{d} \alpha_i K(h_i, \cdot)$. The next stage is to approximate each function h_i by a polynomial of degree N and smaller. Any function $f(h)$ is parameterized by a cross parameter space of coefficients of all d polynomials and α_i. Therefore our proof can inherit many results of (Srinivas et al. 2012). A proof in detail is presented in the supplementary material.

Experiment

We first evaluate the advantages of BFO on a range of applications which are also typical application domains for

BayesOpt: *i*) a synthetic functional optimization problem, *ii*) a hyperparameter optimization problem where we use BFO to choose optimal activation functions of a neural network for the MINIST dataset, and *iii*) policy search in reinforcement learning.

We use RBF kernels $k(x, x') = \exp(-\|x - x'\|/2\sigma_1^2)$, $K(h, h') = \exp(-\|h - h'\|/2\sigma_2^2)$, where σ_1, σ_2 are two hyperparamters. The GP kernel hyperparamets σ_2 is tuned by maximizing the marginal data likelihood.

Functional Optimization: Synthetic Problems

We design two different tasks, $n = 1$ and $n = 2$, of an unknown function $h^* : \Re^n \to \Re$. Each function is a mixture of two (multi-variate) Gaussians, respectively. All optimizers are tasked to minimize

$$J = \int_{x_0}^{x_N} \big(h^*(x) - h(x)\big)^2 dx + \epsilon$$
$$\approx \frac{1}{N} \sum_{i=1}^{N} \big(h^*(x_i) - h(x_i)\big)^2 + \epsilon$$

as a noisy square distance to the unknown function h^*, where $x \in \Re^n, \epsilon \sim \mathcal{N}(0, \sigma^2)$.

We compare its behavior with other base-line methods: standard BayesOpt, RKHS-REMBO (Algorithm 1), and functional gradient descent (assuming to know the true function h^* and ignoring noise).

Functional gradient: Using functional gradient requires to have access to the non-noisy ground-truth function h^* from which $J(h)$ can be approximately evaluated as stated above (without noise ϵ). The functional gradient at h can be computed as $\nabla_h J(h) = \sum_{i=1}^{N} 2\big(h(x_i) - h^*(x_i)\big)K(x_i, \cdot)$ and thus the functional gradient update at iteration l is $h^{\{l+1\}} \leftarrow h^{\{l\}} - \alpha \nabla_h J(h^{\{l\}})$. A sparsification technique (Vincent and Bengio 2002) can be used to achieve a compact representation of h which renders the functional gradient approach an adaptive method too. This means the representation of h will be adaptively adapted to best approximate h^*. Hence, discretization is required to be fine enough to achieve good approximation.

Standard BayesOpt: We assume a parametric representation of h as a linear expansion of N features: $h(x) = \sum_{i=1}^{N} \theta_i \phi_i(x)$. We use RBF features $\phi_i(x) = \exp(-\|x - x_i\|^2/\sigma^2)$ centered around N center points x_i. Standard BayesOpt optimizes over a search space of $\{\theta_i, \{x_i\}_{i=1}^N\}$.

Results: For all optimizers (except the functional gradient method), we use the same number $N = 2$ of features and evaluate them on the two corresponding tasks, $n = 1$ and $n = 2$. The bandwidth σ_1 is set equal to the bandwidth of the Gaussians in the ground-truth function h^*.

We report the mean squared error plot (MSE) J in Fig. 1, together with the final best MSE in Table 1, both averaged over 10 runs. The results show that BFO outperforms all other methods (except the functional gradient which assumes to know the ground-truth). RKHS-REMBO only optimizes on a fixed parameter space in which it can not find a good solution. Standard BayesOpt does not have this problem but is not able to deal with the different scaling of center and weight spaces.

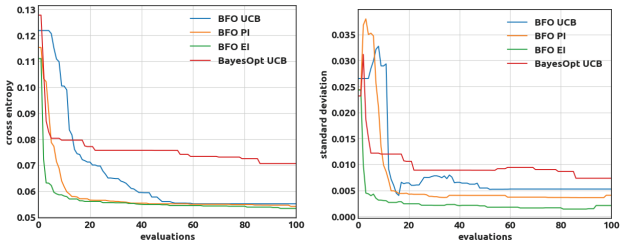


Figure 2: MNIST Dataset: (left) cross-entropy on validation dataset, (right) standard deviations

Table 1: Synthetic domain: MSE and standard deviations of the best evaluation over 10 runs.

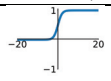
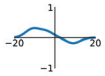
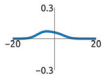
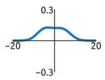
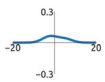
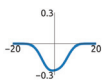
Methods	1D Domain	2D Domain
Functional Gradient	1.31e-6 ± 7.0e-05	2.9e-6 ±7.76e-7
BFO UCB	0.0086 ± 0.0036	0.0085 ±0.002
BFO PI	0.0026 ±0.0025	0.0058 ± 0.002
BFO EI	0.0027 ±0.0014	0.0056 ±0.003
BayesOpt UCB	0.0195 ±0.0086	0.0143 ±0.006
BayesOpt PI	0.0168 ± 0.0088	0.0131 ± 0.002
BayesOpt EI	0.0197 ±0.0070	0.0121 ±0.002
RKHS-REMBO UCB	0.0780 ±4.58e-6	0.0246 ± 0.0001
RKHS-REMBO PI	0.0780 ±1.38e-5	0.0246 ± 7.63e-5
RKHS-REMBO EI	0.0781 ±0.0001	0.0247 ± 0.0001

Hyperparameter Optimization for Neural Networks: Choosing Activation Functions

The MNIST database consists of labeled 28x28 pixel greyscale images of handwritten digits. It contains a test data set of 10.000 data tuples and a training data set of 60.000 data tuples. We train a multilayer perceptron with 2 hidden layers containing 500 and 300 neurons. The network is trained using the cross entropy loss and stochastic minibatch gradient descent with batches of size 100, using TensorFlow with the ADAM optimizer by (Kingma and Ba 2015).

We use three centers for the parametric methods, and also sparsify the functions in BFO to three basis functions (centers in parametric view) to assure a compact representation of the activation functions. We compare BFO to: i) the baseline fixed sigmoid activation function, ii) tunable parametric activation functions (using RBF features), iii) standard BayesOpt using UCB (GP-UCB). We selected the objective functional for Bayesian functional optimization as the cross entropy of the validation data set obtained by training the MLP model with the query activation function. We report the result in Fig. 2, and Table 2. The results clearly show the benefit of our nonparametric BFO approach which outperforms the existing parametric approaches, joint training with tunable activation functions and standard BayesOpt.

Table 2: MNIST Dataset: cross-entropy and standard deviations of the best evaluation over 10 runs.

Methods	CE (val. data)	Test CE (best)	Test Acc.	Activation (best)
Fixed Sigmoid	0.112 ± 0.006	0.097	97.06 %	
Joint Training	0.093 ± 0.007	0.077	97.77%	
BFO UCB	0.055 ± 0.005	0.060	98.06 %	
BFO PI	0.054 ± 0.004	0.058	98.14 %	
BFO EI	0.053 ± 0.002	0.059	98.26%	
GP-UCB	0.070 ± 0.007	0.072	97.59%	

Reinforcement Learning by Policy Search: Inverted Pendulum

For simplicity, we assume a policy π as a Gaussian controller with the mean function $h \in \mathcal{H}$, $a = h(s) = \pi(s)$ where s is a state in the state space $\mathcal{D}$, and a variance σ^2. For parametric policy approaches, h may be a linear function of predefined features as $h(s) = \theta^\top \Phi(s)$, where $\theta \in \Re^N$. For each sample θ, we evaluate $J(\theta) = \mathbb{E}_{\pi(\theta)}\left(\sum_{i=0}^{T} \gamma^i r_i\right)$ which is computed using Monte-Carlo simulations. Specifically, Z trajectories are collected by executing $\pi(\theta)$, and $J(\theta) \approx \frac{1}{Z}\sum_{i=1}^{Z} R(\tau_i)$, where $R(\tau_i)$ is the ith return.

A simple application of BayesOpt for policy search is to define a Euclidean kernel in parameter space (Brochu, Cora, and De Freitas 2010). We compare our direct policy search via BFO using iGP-UCB to standard BayesOpt policy search (Lizotte et al. 2007) (optimizing over a search space of $\{\theta_i\}_{i=1}^N$, while s_i are evenly placed), BayesOpt-A (optimizing over a search space of $\{\theta_i, s_i\}_{i=1}^N$), CMA-ES (Heidrich-Meisner and Igel 2009), a parametric actor-critic, and the actor-critic in RKHS (RKHS-AC) (Lever and Stafford 2015) methods. In all experiments, we use the RBF kernel where the bandwidths are set using the *median-trick*. For the inverted pendulum domain we use the same settings as in (Lever and Stafford 2015). We use $N = 16$ centres, i.e. features, for all algorithms and set discout factor $\gamma = 0.99$ and a horizon $H = 400$.

The results of mean performance and it's 95% confidence are computed over 10 runs and reported in Fig. 3. We observe that all local methods such actor-critic and RKHS actor critic are not competitive as their performance improves too slowly. Also the CMA-ES method is not very data efficient, with a limited number of episodes it's performance remains non-competitive. On contrary, all Bayesian optimization methods, iGP-UCB, GP-UCB, RKHS-REMBO and adaptive BayesOpt, are very competitive. Their performances improve quickly and they exploit data very efficiently.

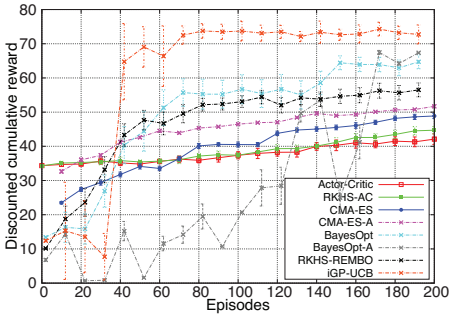


Figure 3: The Inverted Pendulum domain

Conclusion

This paper proposes BFO, a Bayesian functional optimization framework, for global functional optimization. We modeled the function space as a reproducing kernel Hilbert space which results in both, an efficient update of the functional GP and simple optimization of the acquisition functional. Combined with an efficient sparsification method we attain compact and flexible solutions without slowing down the functional GP update too much. Our experiments show that BFO is very promising and able to represent complex solution functions compactly. Compared to other methods BFO can not only theoretically handle functional optimization directly, but also practically does not need to rely on a predefined set of features while bypassing the problem of handling different scales in cross parameter spaces that might occur with standard BayesOpt. We believe that it might be more straightforward and convenient to separately argue about a suited RKHS to represent candidate functions and a functional GP kernel for measuring the similarity between those functions than directly designing a parametric kernel for standard BayesOpt in a parameterized task setting.

References

Auer, P.; Cesa-Bianchi, N.; and Fischer, P. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine Learning* 47(2-3):235–256.

Bagnell, J. A. D., and Schneider, J. 2003. Policy search in reproducing kernel hilbert space. Technical Report CMU-RI-TR-03-45, Robotics Institute, Pittsburgh, PA.

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

Bull, A. D. 2011. Convergence rates of efficient global optimization algorithms. *JMLR* 12(Oct):2879–2904.

Chae, S. B. 1985. *Holomorphy and Calculus in Normed Spaces (Monographs and Textbooks in Pure and Applied Mathematics)*. Marcel Dekker.

Cox, D. D., and John, S. 1992. A statistical method for global optimization. In *Systems, Man and Cybernetics, 1992., IEEE International Conference on*, 1241–1246. IEEE.

Deisenroth, M. P.; Neumann, G.; and Peters, J. 2013. A survey on policy search for robotics. *Foundations and Trends in Robotics* 2(1-2):1–142.

Djolonga, J.; Krause, A.; and Cevher, V. 2013. High-dimensional gaussian process bandits. In *NIPS*, 1025–1033.

Dong, J.; Mukadam, M.; Dellaert, F.; and Boots, B. 2016. Motion planning as probabilistic inference using gaussian processes and factor graphs. In *RSS*.

Heidrich-Meisner, V., and Igel, C. 2009. Neuroevolution strategies for episodic reinforcement learning. *J. Algorithms* 64(4):152–168.

Hennig, P., and Schuler, C. J. 2012. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research* 13(Jun):1809–1837.

Jones, D. R.; Schonlau, M.; and Welch, W. J. 1998. Efficient global optimization of expensive black-box functions. *J. Global Optimization* 13(4):455–492.

Jones, D. R. 2001. A taxonomy of global optimization methods based on response surfaces. *Journal of global optimization* 21(4):345–383.

Jorgensen, P. E. T., and Song, M.-S. 2007. Entropy encoding, hilbert space, and karhunen-love transforms. *Journal of Mathematical Physics* 48.

Kadri, H.; Duflos, E.; Preux, P.; Canu, S.; Rakotomamonjy, A.; and Audiffren, J. 2015. Operator-valued kernels for learning from functional response data. *Journal of Machine Learning Research* 16:1–54.

Kasim, M. F., and Norreys, P. A. 2016. Infinite dimensional optimistic optimisation with applications on physical systems. *arXiv preprint arXiv:1611.05845*.

Kingma, D., and Ba, J. 2015. Adam: A method for stochastic optimization. In *ICLR*.

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

Lever, G., and Stafford, R. 2015. Modelling policies in mdps in reproducing kernel hilbert space. In *AISTATS*.

Lizotte, D. J.; Wang, T.; Bowling, M. H.; and Schuurmans, D. 2007. Automatic gait optimization with gaussian process regression. In *IJCAI*, 944–949.

Lizotte, D. J. 2008. *Practical Bayesian Optimization*. Ph.D. Dissertation, University of Alberta, Edmonton, Alberta, Canada.

Marinho, Z.; Boots, B.; Dragan, A. D.; Byravan, A.; Gordon, G. J.; and Srinivasa, S. 2016. Functional gradient motion planning in reproducing kernel hilbert spaces. In *RSS*.

Močkus, J. 1975. *On bayesian methods for seeking the extremum*. Springer. 400–404.

Munos, R. 2011. Optimistic optimization of a deterministic function without the knowledge of its smoothness. In *NIPS*, 783–791.

Pandey, S.; Chakrabarti, D.; and Agarwal, D. 2007. Multi-armed bandit problems with dependent arms. In *ICML*, 721–728.

Rasmussen, C. E. 2006. *Gaussian processes for machine learning*. MIT Press.

Shahriari, B.; Wang, Z.; Hoffman, M. W.; Bouchard-Côté, A.; and de Freitas, N. 2014. An entropy search portfolio for bayesian optimization. *arXiv preprint arXiv:1406.4625*.

Shi, J. Q., and Choi, T. 2011. *Gaussian process regression analysis for functional data*. CRC Press Boca Raton, FL:.

Srinivas, N.; Krause, A.; Kakade, S. M.; and Seeger, M. W. 2012. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE Trans. Information Theory* 58(5):3250–3265.

Tyagi, H.; Kyrillidis, A.; Gärtner, B.; and Krause, A. 2016. Learning sparse additive models with interactions in high dimensions. In *AISTATS*, 111–120.

Tyagi, H.; Gärtner, B.; and Krause, A. 2014. Efficient sampling for learning sparse additive models in high dimensions. In *NIPS*, 514–522.

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

Vien, N. A.; Dang, V.-H.; and Chung, T. 2017. A covariance matrix adaptation evolution strategy for direct policy search in reproducing kernel hilbert space. In *ACML*.

Vien, N. A.; Englert, P.; and Toussaint, M. 2016. Policy search in reproducing kernel hilbert space. In *IJCAI*, 2089–2096.

Vincent, P., and Bengio, Y. 2002. Kernel matching pursuit. *Machine Learning* 48(1-3):165–187.

Wang, Z.; Hutter, F.; Zoghi, M.; Matheson, D.; and de Freitas, N. 2016. Bayesian optimization in a billion dimensions via random embeddings. *JAIR* 55:361–387.

Wilson, A. G. 2014. *Covariance kernels for fast automatic pattern discovery and extrapolation with Gaussian processes*. Ph.D. Dissertation, Univ. of Cambridge, UK.