An experimental study in adaptive kernel selection for Bayesian Optimization

Ibai Roman, Roberto Santana,
Alexander Mendiburu, Jose A. Lozano

January 2, 2020

Abstract

Bayesian Optimization has been widely used along with Gaussian Processes for solving expensive-to-evaluate black-box optimization problems. Overall, this approach has shown good results, and particularly for parameter tuning of machine learning algorithms. Nonetheless, Bayesian Optimization has to be also configured to achieve the best possible performance, being the selection of the kernel function a crucial choice. This paper investigates the convenience of adaptively changing the kernel function during the optimization process, instead of fixing it a priori. Six adaptive kernel selection strategies are introduced and tested in well-known synthetic and real-world optimization problems. In order to provide a more complete evaluation of the proposed kernel selection variants, two major kernel parameter setting approaches have been tested. According to our results, apart from having the advantage of removing the selection of the kernel out of the equation, adaptive kernel selection criteria show a better performance than fixed-kernel approaches.

1 Introduction

IN many machine learning algorithms, parameters need to be fine-tuned in order to guarantee good performance. Each parameter, discrete or continuous, influences the overall behavior of the algorithm. Thus, some domain knowledge is needed to choose a good parameter set. However, when there is no human expertise, a trial and error strategy may not be the most efficient method to find a suitable set of parameters. Moreover, some machine learning processes can take a long time to complete, and manual optimization might be intractable.

Finding the best parameter set by executing the machine learning algorithm and observing the result can be seen as a nonlinear function optimization problem. It can be considered that each parameter set is a point or solution ($\mathbf{x}$) in the search space, and the error of the learning process is the outcome of the objective function ($f(\mathbf{x})$). The best parameter set ($\mathbf{x}^*$) is the one that minimizes the error, and can be mathematically expressed as follows:

$$\mathbf{x}^* = argmin_{\mathbf{x} \in A \subset \mathbb{R}^{n_d}} f(\mathbf{x}) \tag{1}$$

where $\mathbf{x}$ is a vector of n_d variables whose components can take values in $\mathbb{R}$ or $\mathbb{Z}$. Note that parameters are bounded, and so is the search space (A).

This optimization problem has some key properties. First, the objective function is a black-box function as its analytic form is unknown [1]. This property reduces the number of algorithms that can be used to solve this problem. Second, the objective function is frequently very expensive to evaluate. In consequence, it is desirable to achieve the optimum value with as few function evaluations as possible.

Bayesian Optimization (BO) [2] is a state-of-the-art global optimization technique suitable for this kind of optimization problems [3, 4]. BO is a sequential optimization algorithm, where the sampling strategy is based on a probability distribution over all the possible objective functions. This probability distribution acts as a surrogate model, and it is updated every time a solution is evaluated using the objective function.

Gaussian Processes (GPs) are a popular choice as surrogate models in BO [5–12]. A GP is a collection of random variables of which any finite set will have a joint Gaussian distribution [13]. Note that, if we reduce this joint distribution to one dimension, it will represent a Gaussian distribution over the outcome of the objective function for a particular solution in the search space.

A GP is completely defined by a mean function and a covariance function. This covariance function establishes the relation between the objective function values of each pair of solutions by means of a kernel. These kernels usually depend on some parameters which also influence the covariance function. Hence, the election of the kernel and its parameters is crucial for BO performance when using the GPs as the surrogate model. Since manual election of these configurations is not an efficient option, it seems reasonable to automatically select them to avoid another parameter optimization problem.

Although kernel parameter tuning has attracted much attention, kernel selection has not been extensively studied in the BO research field. In most applications, they are selected in advance by an expert [14–16] or a search algorithm [17], according to the optimization problem.

In this paper we present a general framework for adaptive kernel selection in BO. We evaluate different approaches that adaptively combine and select information from multiple kernels to guide the search. For this purpose, these methods will require managing several GPs in parallel through the whole search process, each one with a different kernel, which increases the computation cost. However, we assume this extra computation time will be negligible compared to the time needed to compute an expensive-to-evaluate objective function.

Previous approaches have investigated ways to eliminate the kernel selection from the initial BO configuration. For example, [18] also used several GPs in parallel with different kernels. Instead of choosing one of them at each step, a mixture of kernels was used in order to find the next point to sample.

Similar strategies have been proposed in portfolio allocation approaches [19, 20] to solve the acquisition function selection problem. In this work we deal with the kernel selection problems instead.

Other approaches have also tried to manage several surrogate models apart from the GPs in parallel. [21] carried out an optimization process with multiple surrogate models, such as radial basis neural networks, linear Shepard, and several support vector regression methods. All the solutions proposed by these surrogate models were evaluated in parallel. In this work, a variant of this method will be proposed, using GPs with different kernel functions as surrogate models.

The objective of this work is threefold: First, to demonstrate the influence of the kernel in the performance of the BO algorithm. Also, to introduce new additional proposals for adaptive kernel selection. And finally, to validate all the adaptive methods, comparing them against fixed-kernel approaches.

The remainder of the paper is structured as follows: In Section 2, a background on BO and GPs is provided. Section 3 describes the kernel selection criteria. In Section 4, the experimental setup is presented, and the results are discussed in Section 5. Finally, in Section 6, the conclusions and the future work are presented.

2 Background

2.1 Bayesian Optimization

Bayesian Optimization (BO) [2] is a sequential optimization algorithm suitable for black-box functions. BO uses a probability distribution over all the possible objective functions to determine the sampling strategy. As the analytic form of the objective function is generally unknown, BO treats it like a random function, and places a prior belief about the space of possible objective functions. Every time the objective function is evaluated, this prior belief is updated with the likelihood of having those observations, generating a posterior distribution over functions. This updating process is based on the Bayes theorem, where the *posterior* probability of a model given some evidence is proportional to the *prior* probability of the model multiplied by the *likelihood* of the evidence given the model. This can be expressed as:

$$P\left(f|D_{1:t}\right) \propto P\left(D_{1:t}|f\right) P\left(f\right) \tag{2}$$

where $D_{1:t}$ is the data set of solutions and their respective objective function evaluations $\{(\mathbf{x}_i, f(\mathbf{x}_i))\}_{i=1}^{t}$.

The sampling strategy will use a utility function ($u(\cdot)$), often called acquisition function, that provides a measure of the utility of each solution. In order to provide this measure, the acquisition function relies on the probability distribution over all the possible objective functions. This probability distribution acts as a surrogate model (SM).

Algorithm 1 illustrates the whole process of the BO algorithm [22].

Algorithm 1 BO algorithm

```
1: procedure BAYESIAN-OPTIMIZATION($\mathbf{x}_1$)
2:     $y_1 = f(\mathbf{x}_1)$                                  ▷ Sample a random point
3:     $D_{1:1} = \{(\mathbf{x}_1, y_1)\}$                        ▷ Initialize the data set
4:     $SM \leftarrow D_{1:1}$                                 ▷ Initialize the surrogate model
5:     $t = 2$
6:     repeat
7:         $\mathbf{x}_t = \arg\max_{\mathbf{x} \in A}\ u(\mathbf{x}|SM)$      ▷ Select the next point to evaluate
8:         $y_t = f(\mathbf{x}_t)$                             ▷ Evaluate the objective function
9:         $D_{1:t} = D_{1:t-1} \cup \{(\mathbf{x}_t, y_t)\}$          ▷ Update the data set
10:        $SM \leftarrow D_{1:t}$                              ▷ Update the surrogate model
11:        $t = t + 1$
12:    until the stopping criterion is met
13: end procedure
```

The optimization process begins with the sampling of a random solution and the initialization of the *SM*. Then, the point that maximizes the expected utility is selected for sampling. Finally, the data set is augmented with the result of this evaluation and the *SM* is updated accordingly. This process is repeated until the stopping criterion is met.

In the following sections we will show how this acquisition function is generated from the posterior distribution, when using GPs as *SM* in BO.

2.2 Gaussian Processes

*SM*s are a key element in Algorithm 1, as the acquisition function will rely on them to select the next point. One of the most popular choices in BO is to use a Gaussian Process (GP) as *SM*. A GP is a stochastic process, defined by a collection of random variables, any finite number of which has a multivariate Gaussian distribution [13]. The key property that makes GPs convenient to BO, is that, a GP is conjugate to itself with respect to a Gaussian likelihood function. Thus, after the sampling of several points of the objective function, the a posteriori distribution over functions remains a GP.

As previously mentioned, GPs can be completely defined by a mean function ($m(\mathbf{x})$) and a covariance function, which depends on a kernel ($k(\mathbf{x}, \mathbf{x}')$). Given that, the GP can be expressed as follows:

$$f(x) \sim GP(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \tag{3}$$

Usually, a non-informative mean function is used, such as $m(\mathbf{x}) = 0$. However, in more sophisticated approaches, this function may depend on $\mathbf{x}$.

The kernel establishes the covariance between the objective function values of two different points. In most common kernels used in BO, this covariance depends on the distance between the points, given by the following translation invariant distance measure:

$$k(\mathbf{x}, \mathbf{x}') = \hat{k}(r) \text{ where } r = \sqrt{\sum_{d=1}^{n_d} \left(\frac{x_d - x'_d}{\theta_{l_d}} \right)^2} \tag{4}$$

where θ_{l_d} is the length-scale parameter for each dimension d and it expresses the relevance of each dimension in the output.

Many kernels have been proposed in the literature, being probably the square exponential kernel the most popular one. This kernel is expressed as follows:

$$k_{SE}(r) = \theta_0^2 \ exp \left(-\frac{1}{2} r^2 \right) + \theta_n \tag{5}$$

where θ_0 and θ_n are the amplitude and noise parameters respectively. The amplitude is a length-scale parameter for all dimensions, while the noise parameter allows the GP to adapt the random function when the observations of the objective function are noisy [23]. Similar to the square exponential, most kernels depend on certain parameters that will be denoted as $\mathbf{\Theta}$. In the previous case $\mathbf{\Theta} = (\theta_0, \boldsymbol{\theta}_l, \theta_n)$.

2.3 Acquisition function

Once we have a way to approximate the value of the objective function through the SM, the acquisition function is placed to select the next point. It assigns a measure of utility for each point in the search space given the *SM*. This measure of utility balances the exploration versus exploitation trade-off.

There are several acquisition functions, such as probability of improvement [24], expected improvement [25] or GP-UCB [26]. From these, we have chosen expected improvement (u_{EI}), as it has a competitive performance in BO [27] and it does not need parameters to configure. This acquisition function measures the expectation of improving the best result:

$$\begin{gathered} u_{EI}(\mathbf{x}) = \begin{cases} \sigma(\mathbf{x}) \left(Z \cdot \Phi(Z) + \phi(Z) \right) & \text{if } \sigma(\mathbf{x}) > 0 \\ 0 & \text{if } \sigma(\mathbf{x}) = 0 \end{cases} \\ where \\ Z = \begin{cases} \frac{f(\mathbf{x}^+) - \mu(\mathbf{x})}{\sigma(\mathbf{x})} & \text{if } \sigma(\mathbf{x}) > 0 \\ 0 & \text{if } \sigma(\mathbf{x}) = 0 \end{cases} \end{gathered} \tag{6}$$

where $\mathbf{x}^+$ is the best point so far. Φ denotes the cumulative distribution function (CDF) and ϕ the probability density function (PDF) of the standard normal distribution (with 0 mean and a variance of 1).

In the previous equation $\mu(\mathbf{x})$ and $\sigma(\mathbf{x})$ represent the expected value for the objective function at point $\mathbf{x}$ and its variance. At step $t + 1$, these values are given by the following equation:

$$\begin{aligned}\mu(\mathbf{x}) &= m(\mathbf{x}) + k(\mathbf{X}_{1:t}, \mathbf{x})k(\mathbf{X}_{1:t}, \mathbf{X}_{1:t})^{-1} f(\mathbf{X}_{1:t}) \\ \sigma^2(\mathbf{x}) &= k(\mathbf{x}, \mathbf{x}) - k(\mathbf{X}_{1:t}, \mathbf{x})k(\mathbf{X}_{1:t}, \mathbf{X}_{1:t})^{-1} k(\mathbf{x}, \mathbf{X}_{1:t})\end{aligned} \tag{7}$$

where $k(\mathbf{X}_{1:t}, \mathbf{x})$ and $k(\mathbf{x}, \mathbf{X}_{1:t})$ are vectors with the value of the kernel function between $\mathbf{x}$ and all the previously evaluated solutions. $k(\mathbf{X}_{1:t}, \mathbf{X}_{1:t})$ is the matrix of the kernel values between all the pairs of the previously evaluated solutions.

3 Adaptive Kernel Selection Criteria

As discussed in Section 1, kernel parameter tuning has been already addressed in the literature and efficient methods have been proposed for it [28, 29]. However, even when kernels do play a main role in the performance of BO, kernel selection has not attracted too much interest. We propose to carry out an optimization process with several GPs in parallel (each one with a different kernel) and adaptively choosing one of them. By adaptively choosing the kernel during the optimization process, there is no need to select a kernel in advance. Moreover, this strategy is expected to improve the search results of fixed-kernel approaches, as the application of many diverse models may mitigate over-fitting. In this section, seven different methods for adaptive kernel selection will be proposed and discussed.

Our algorithm resembles Algorithm 1 with two changes. First (see Algorithm 2), there will be several independent GPs working in parallel, one per kernel function. At each step t, each GP k will suggest one solution $(\mathbf{x}_{t,k})$ by optimizing the corresponding acquisition function. Second, a *choose* function is defined in order to select the next point to evaluate $(\mathbf{x}_{t,k^+})$ between all the points proposed by the GPs. Different criteria can be implemented in the *choose* function, leading to different BO behaviors. Finally, the data set, $D_{1:t}$, will be augmented with the selected solution, and all the GPs will be updated accordingly. Note that all the GPs keep the same data set. As there is one kernel per GP, from now on, we will refer to these surrogates as either kernels or GPs. Algorithm 2 describes these modifications, where n_s refers to the number of GPs. The whole process is also illustrated in Figure 1.

The computational cost of Algorithm 2 is linked to the update of the GPs, since the hyperparameters of each kernel must be adapted every time a solution is added to the data. Thus, the number of GPs indicates the complexity added to Algorithm 1. Nevertheless, in those cases where the evaluation of the objective function is very expensive, the relative cost of both algorithms is negligible.

In this work, we propose six new strategies to implement the *choose* function, namely *Random Uniform*, *Best Likelihood*, *Best Utility*, *Weighted Best*, *Parallel Test* and *Utility Mean*. Additionally, we include the approach proposed by [18], which we will refer as *Weighted Mixture*.

Algorithm 2 Adaptive Kernel Selection algorithm

```
1: procedure ADAPTIVE-KERNEL-SELECTION($\mathbf{x}_1$)
2:     $y_1 = f(\mathbf{x}_1)$                                   ▷ Sample a random point
3:     $D_{1:1} = \{(\mathbf{x}_1, y_1)\}$                          ▷ Initialize the data set
4:     for $k = 1$ to $n_s$ do                                   ▷ Initialize the $GP$s
5:         $GP_{t,k} \leftarrow D_{1:1}$
6:     end for
7:     t = 2
8:     repeat
9:         for $k = 1$ to $n_s$ do                               ▷ Each GP suggests a sol.
10:            $\mathbf{x}_{t,k} = \arg\max_{\mathbf{x}\in A}\ u(\mathbf{x}|GP_{t,k})$
11:        end for
12:        $k^+ = choose(1 : n_s)$                               ▷ Choose a solution
13:        $\mathbf{x}_t = \mathbf{x}_{t,k^+}$
14:        $y_t = f(\mathbf{x}_t)$                                  ▷ Evaluate the objective function
15:        $D_{1:t} = D_{1:t-1} \cup \{(\mathbf{x}_t, y_t)\}$            ▷ Update the data set
16:        for $k = 1$ to $n_s$ do                               ▷ Update the $GP$s
17:            $GP_{t,k} \leftarrow D_{1:t}$
18:        end for
19:        t = t+1
20:    until the stopping criterion is met
21: end procedure
```

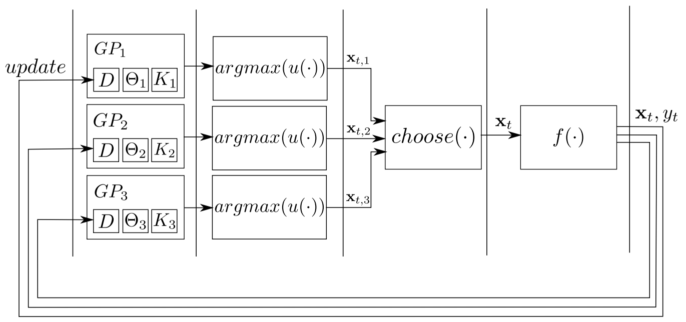


Figure 1: Adaptive kernel selection diagram. Although the main loop of the algorithm remains the same, several acquisition functions are optimized simultaneously, each one related to a different kernel. Then, according to the *choose* function, the best point is selected, evaluated and added to the data set.

3.1 Random Uniform

Random Uniform is the simplest strategy. At each step t, a GP is randomly selected. The next point to evaluate ($\mathbf{x}_{t,k^+}$) will be proposed by the acquisition function using the selected GP:

$$k^+ \sim U\{1, n_s\} \tag{8}$$

In this algorithm, important performance gains can be achieved if the GP that suggests the next solution to sample is selected beforehand, since it is only necessary to update that GP.

3.2 Best Likelihood

This approach relies on the Gaussian likelihood function ($\mathcal{L}$). After the adaptation of the kernel parameters, each GP will have a different likelihood value given the data. The next point to evaluate will be selected according to the GP with the highest likelihood value:

$$k^+ = \arg\max_k \mathcal{L}(GP_{t,k}) \tag{9}$$

3.3 Best Utility

Based on the acquisition function, the expectation of improvement of each GP is compared. All the acquisition functions are optimized, one per GP, and each one proposes a new point to evaluate ($\mathbf{x}_{t,k}$). The point with the highest utility ($\mathbf{x}_{t,k^+}$) is selected for the next evaluation:

$$k^+ = \arg\max_k u(\mathbf{x}_{t,k}|GP_{t,k}) \tag{10}$$

3.4 Weighted Best

The *Best Utility* criterion selects the most optimistic GP. However, it may happen that some GPs promise good utility values and, when evaluated, the improvement is not as good as expected. When this behavior is repeated, it may lead to poor results. Thus, a weighting system is introduced to reduce the influence of optimistic kernels:

$$k^+ = \arg\max_k w_{t,k} u(\mathbf{x}_{t,k}|GP_{t,k}) \tag{11}$$

where $w_{k,t}$ is the weight assigned to $GP_{t,k}$ at evaluation t.

Initially, all weights are set to 0.5. Every evaluation, the weight of the GP that proposed the evaluated point is updated, considering the relative improvement produced with respect to the best result so far as shown in Equation (12). The CDF of the standard normal distribution (Φ) is used to limit the update rate of the weights between $[0.5, 1.5]$. Thus, the kernels that are penalized are still able to improve their weights if they provide good predictions.

$$w_{t+1,k} = \begin{cases} w_{t,k}\left(\Phi\left(\frac{f(\mathbf{x}^+)-f(\mathbf{x}_{t,k})}{|f(\mathbf{x}^+)|}\right)+0.5\right) & \text{if } k = k_t^+ \\ w_{t,k} & \text{otherwise} \end{cases} \tag{12}$$

where $\mathbf{x}^+$ is the best point so far.

3.5 Parallel Test

This method is based on the approach proposed by [21], using GPs with different kernel functions as surrogate models. Instead of selecting only one solution that is expected produce the highest improvement, in this approach all the solutions are selected for evaluation. Then, all the GPs are updated with the results of those evaluations.

If the objective function is expensive to sample but it can be evaluated in parallel with no additional cost, this algorithm produces major savings in terms of computational cost.

3.6 Utility Mean

In this method a mixture of acquisition functions is used to choose the next point to sample. We implement this approach assigning the same weight $(1/n_s)$ to each acquisition function, i.e. computing a mean utility. Note that, in this case, the loop in line 9 of Algorithm 2 is no longer needed, as the acquisition functions are all maximized together. The next point to evaluate will be the one that maximizes the following function:

$$\mathbf{x}_{t,k^+} = \arg\max_{\mathbf{x}\in A}\left(\frac{1}{n_s}\sum_{k=1}^{n_s} u(\mathbf{x}|GP_{t,k})\right) \tag{13}$$

3.7 Weighted Mixture

Similar to the previous approach, a mixture of kernels is used to choose the next point to sample. However, as suggested by [18], in this approach the acquisition functions are weighted by their likelihood ratios:

$$\begin{aligned} \mathbf{x}_{t,k^+} &= \arg\max_{\mathbf{x}\in A}\left(\sum_{k=1}^{n_s} w_{t,k}u(\mathbf{x}|GP_{t,k})\right) \\ &\textit{where} \\ w_{t,k} &= \frac{\mathcal{L}(GP_{t,k})}{\sum_{i=1}^{n_s}\mathcal{L}(GP_{t,i})} \end{aligned} \tag{14}$$

4 Experimental setup

The objective of the experimentation is to validate the adaptive kernel selection criteria shown in the previous section. They will be compared against fixed-kernel approaches in six synthetic and real-world optimization problems.

The optimization strategies are divided into two groups, the adaptive strategies and the fixed-kernel strategies. Each adaptive optimization strategy will follow Algorithm 2, with one of the *choose* functions explained in Section 3. Fixed-kernel optimization strategies will not change the kernel function during the optimization process. They will follow Algorithm 1, each with a different kernel. In addition, a fixed-kernel optimization strategy is introduced, called *FixedAtRandom*, which simulates a random kernel selection at the beginning of the BO run.

Table 1 shows the six well-known kernels [13] used in the experiments.

Kernel function expressions	
Exponential	$k_E(r) = \theta_0^2 \; exp\,(-r) + \theta_n$
γ-exponential	$k_{E\gamma}(r) = \theta_0^2 \; exp\,(-r^\gamma) + \theta_n$
Squared Exp.	$k_{SE}(r) = \theta_0^2 \; exp\left(-\frac{1}{2}r^2\right) + \theta_n$
Matern 32	$k_{M32}(r) = \theta_0^2 \left(1 + \sqrt{3}r\right) exp\left(-\sqrt{3}r\right) + \theta_n$
Matern 52	$k_{M52}(r) = \theta_0^2 \left(1 + \sqrt{5}r + \frac{5}{3}r^2\right) exp\left(-\sqrt{5}r\right) + \theta_n$
Rat. Quadratic	$k_{RQ}(r) = \theta_0^2 \left(1 + \frac{1}{2\alpha}r^2\right)^{-\alpha} + \theta_n$

Table 1: Well-known kernel functions. θ_0 and θ_n are the kernel hyperparameters called amplitude and noise respectively. In the γ-exponential kernel $\gamma = 1.5$ was set and in the Rational Quadratic kernel $\alpha = 2$.

Experiments have been conducted on six optimization problems included in the *HPOlib* [30] parameter optimization library. These optimization problems, shown in Table 2, have been optimized using all the previously described adaptive and fixed-kernel strategies. Each optimization process was carried out until the maximum number of function evaluations (*Max. Eval.*) was reached. At every function evaluation, the best result so far ($f(\mathbf{x}^+)$) is recorded. Due to the random choice of the first solution and the stochastic nature of the kernel parameter setting methods, each experiment was repeated 30 times.

The proposed *choose* functions were implemented as part of a software framework, called *BOlib*[1], specifically developed for this work. Similar to *Hyperopt* [31], *DiceKriging* R package [32], *Spearmint* [33], *BayesOpt* [34] and *PyBO* [35], it implements the most common BO variants. Moreover, it includes the *choose* functions introduced in Section 3.

To validate our framework, in a first step *BOlib* was compared to *Spearmint* [33], a reference software for BO-GP. The same BO configuration proposed by the authors of Spearmint was used in a preliminary experimentation, and the

[1]https://pypi.org/project/bolib/

Benchmarks	Params	Dataset	Max. Eval.
Branin-Hoo	2 continuous	- (synthetic)	200
Hartmann 6D	6 continuous	- (synthetic)	200
LDA on grid	3 discrete	wikipedia articles	50
SVM on grid	3 discrete	UniPROBE	100
Log. Reg. (no cv)	4 continuous	MNIST	100
Log. Reg. (cv)	4 continuous	MNIST	100

Table 2: Optimization problems used as benchmarks.

results showed no statistically significant differences between both implementations.

The experimentation shown in this work, uses the following BO configuration. First of all, the expected improvement is used as the acquisition function. Regarding GPs, a constant mean function is used $m(\mathbf{x}) = \theta_\mu$, where θ_μ is selected along with the kernel parameters $\boldsymbol{\Theta}$. For this kernel parameter selection, the most common approaches are the optimization of the likelihood function (OPT), and the Markov Chain Monte Carlo approach (MCMC) described by [29]. We run our experiments using both techniques OPT and MCMC. Finally, to maximize the acquisition function, candidate points are sampled from a low discrepancy Sobol sequence [36].

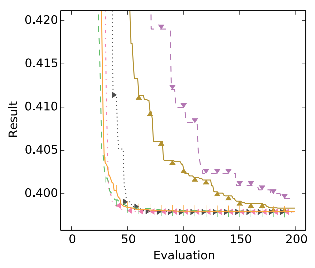
5 Results

The experimentation was performed in two steps according to the objectives of this paper. First, an initial set of experiments was performed with fixed-kernel strategies to confirm the influence of the kernel in the performance of the BO algorithm. Second, adaptive kernel selection strategies were added and compared to the former ones.

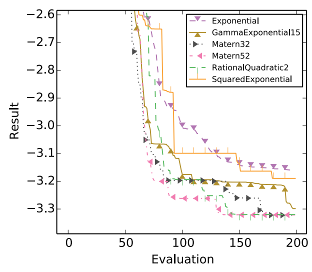
5.1 Kernel influence

As previously mentioned, and pointed out in the literature, kernel selection is a key aspect in BO [13]. Hence, before we start introducing the adaptive selection strategies, fixed-kernel strategies were tested. In this first experiment only the OPT kernel parameter optimization method was used. Recall that these fixed-kernel strategies have the same BO configuration, except for the particular parameters of each kernel. They were tested in all the optimization problems introduced in Section 4. The best result so far was recorded for every function evaluation.

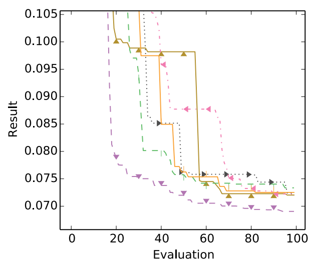
Figure 2 shows the results of this first experiment. As each optimization process with the same BO configuration was repeated 30 times, median values are shown for each one.



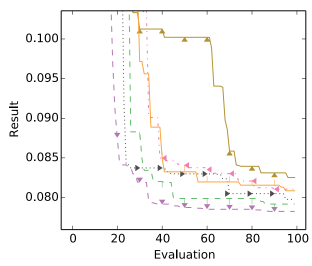
(a) Branin-Hoo



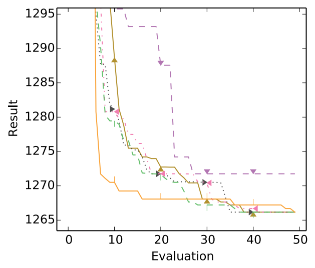
(b) Hartmann 6D



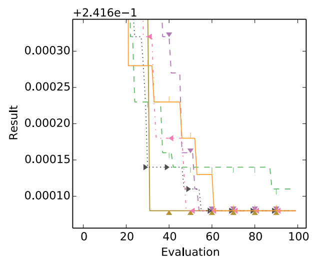
(c) Log. Regression (no cv)



(d) Log. Regression (cv)



(e) LDA on grid



(f) SVM on grid

Figure 2: At each evaluation step, the best objective function value found is shown. Each colored line illustrates the median value of all runs for each Kernel.

According to this figure, there is no kernel that outperforms the rest in all the problems. Apart from the differences in the final result, the results also differ in their evolution throughout the process. The kernels that show an exploratory behavior in some problems, tend to be more oriented to exploitation in others. For example, the *Exponential* kernel achieves good results in *Logistic Regression* problems from early stages of the optimization, while in *Branin-Hoo* and *LDA on grid* problems shows the worst behavior among all the kernels in any stage of the search. In summary, it can be said that the kernel highly influences the BO dynamic, and that there is no rule-of-thumb to choose the most appropriate kernel for each problem.

5.2 Adaptive/Fixed comparison

As choosing the best performing kernel for a particular problem is not an easy task, the adaptive kernel selection strategies described in Section 3 are introduced and compared to the fixed-kernel approaches. The best result per run was recorded as performance metric for each strategy.

First, an OPT/MCMC comparison was made. Both kernel parameter selection approaches were tested for each optimization strategy, one against the other, using Wilcoxon test [37] ($\alpha = 0.05$). For the sake of clarity, we performed the following tests with OPT approaches only, taking into account that the results obtained with these methods were better than those achieved with methods using MCMC. Then, in order to analyze if there are significant differences between the strategies, a Kruskal-Wallis test (K-W) [38] was carried out ($\alpha = 0.05$). Finally, in problems where significant differences were achieved, post-hoc tests were performed. Particularly, SCMAMP R package [39] was used to apply the Wilcoxon signed rank test as in [37], while p-values were corrected using Shaffer's (static) procedure, as in [40].

In Table 3 all the optimization strategies are shown, ordered by the average results achieved in each optimization problem. Overall, adaptive strategies show a better performance than fixed-kernel approaches. Particularly, they occupy the highest (best) positions in the ranking for 5 out of 6 problems (although in *SVM on grid* no statistical differences were found). In general terms, *Parallel Test* seems to be the best performing technique, as it outperforms the rest in *Branin-Hoo* and *Log. Regression (cv)* problems and it is the second best in *Hartmann 6D* and *SVM on grid*. We found that the variance in the results of the proposed approaches is similar to the fixed-kernel methods. Therefore, the sensitivity of the initial conditions for the proposed algorithm is comparable to the traditional BO.

Regarding the kernel parameter selection strategies, in most cases, there are no significant differences. However, in the synthetic approaches some of the optimization strategies perform better using the MCMC approach, while in the real-world problems, better results were achieved using OPT.

6 Conclusions and Future Work

While BO along with GPs is acknowledged to be an efficient approach to deal with optimization problems with a limited budget of function evaluations, the question of kernel selection remains one of its main limitations. This question is particularly critical for machine learning parameter selection problems, for which the manual selection of parameters is not an affordable option in many real-world applications. In this paper we have proposed and analyzed different strategies for adaptive kernel selection, showing, based on an extensive experimentation and a statistical analysis of the results, that automatic selection

Branin-Hoo*			Hartmann 6D*			Log. Regression (no cv)*		
Score	Strategy	Comp.	Score	Strategy	Comp.	Score	Strategy	Comp.
10	**ParallelTest**	**-**	**10**	**RandomUniform**	**MCMC**	1	Exponential	OPT
10	Matern32	-	**9**	**ParallelTest**	**MCMC**	**0**	**BestUtility**	**OPT**
10	**RandomUniform**	**MCMC**	**7**	**WeightedMixture**	**MCMC**	**0**	**WeightedMixture**	**OPT**
8	**WeightedBest**	**-**	**5**	**UtilityMean**	**-**	**0**	**WeightedBest**	**OPT**
7	Matern52	MCMC	**3**	**BestUtility**	**-**	**0**	**ParallelTest**	**OPT**
6	SquaredExponential	-	**3**	**WeightedBest**	**MCMC**	**0**	**RandomUniform**	**-**
6	RationalQuadratic2	-	3	RationalQuadratic2	-	**0**	**UtilityMean**	**OPT**
5	FixedAtRandom	-	2	GammaExponential15	-	0	GammaExponential15	-
3	GammaExponential15	-	1	Matern32	MCMC	0	RationalQuadratic2	-
3	**UtilityMean**	**MCMC**	1	Matern52	MCMC	0	FixedAtRandom	-
0	Exponential	-	0	FixedAtRandom	MCMC	0	Matern52	-
0	**WeightedMixture**	**-**	0	Exponential	-	0	SquaredExponential	-
0	**BestUtility**	**MCMC**	0	SquaredExponential	MCMC	0	Matern32	-
0	**BestLikelihood**	**-**	**0**	**BestLikelihood**	**-**	**0**	**BestLikelihood**	**-**
Log. Regression (cv)*			LDA on grid*			SVM on grid		
Score	Strategy	Comp.	Score	Strategy	Comp.	Score	Strategy	Comp.
3	**ParallelTest**	**OPT**	**0**	**UtilityMean**	**OPT**	**0**	**WeightedBest**	**OPT**
1	**BestUtility**	**OPT**	**0**	**WeightedMixture**	**OPT**	**0**	**ParallelTest**	**-**
1	**WeightedBest**	**OPT**	0	Matern32	-	**0**	**BestUtility**	**-**
0	Exponential	OPT	0	FixedAtRandom	-	0	Exponential	OPT
0	**RandomUniform**	**OPT**	**0**	**RandomUniform**	**-**	**0**	**RandomUniform**	**-**
0	RationalQuadratic2	-	**0**	**ParallelTest**	**OPT**	0	Matern52	OPT
0	Matern52	-	0	RationalQuadratic2	-	**0**	**BestLikelihood**	**OPT**
0	FixedAtRandom	-	0	Matern52	-	0	Matern32	-
0	**UtilityMean**	**OPT**	0	SquaredExponential	-	**0**	**WeightedMixture**	**-**
0	**BestLikelihood**	**-**	0	GammaExponential15	-	0	GammaExponential15	OPT
0	Matern32	-	**0**	**BestUtility**	**-**	**0**	**UtilityMean**	**-**
0	**WeightedMixture**	**-**	**0**	**WeightedBest**	**-**	0	FixedAtRandom	-
0	SquaredExponential	-	**0**	**BestLikelihood**	**-**	0	SquaredExponential	-
0	GammaExponential15	-	0	Exponential	-	0	RationalQuadratic2	-

Table 3: Results of the statistical tests. The asterisk next to the problem title indicates that statistical differences where found in the K-W test. The left column shows the score for the OPT configurations per optimization problem. This score denotes the number of times that the indicated configuration is significantly better than other configurations. In the middle, the names of the strategies are shown, where the adaptive ones are represented in bold. The results of the OPT/MCMC comparison are shown in the right most column.

of kernels is indeed a more efficient approach when considering optimization problems with different characteristics.

This work has made the following contributions:

- We have introduced six new adaptive kernel selection criteria (presented in Section 3) that implement different ways of using the information about the kernels for selecting the points to be sampled.
- An extensive evaluation of the introduced algorithms has been made, including a comparison with BO that uses fixed kernels on six commonly used benchmark problems, and the analysis of the results has been supported by rigorous statistical tests.
- Our research has provided insights about the difference between BO methods when using the optimization of the likelihood function (OPT) and the MCMC approach for kernel parameter selection. Our results show that in most real-world scenarios OPT is still the best choice for BO.

- A modular software framework, called *BOlib*, has been implemented. This software incorporates some of the best features of similar implementations (e.g. *HPOlib*) and implements the adaptive methods investigated in the paper. *BOlib* can be easily extended to include other optimization problems and functionalities.

Our main message is that the use of adaptive techniques can be incorporated to BO in a straightforward way to obtain a more robust behavior than BO with fixed kernels along diverse problems. When considering the application of BO in the context of machine learning methods that can be applied to problems with different characteristics, the inclusion of adaptive methods may be essential. Regarding the particular choice of the adaptive strategy, we have shown that *Parallel Test* produces good results.

In spite of the achievements, some further research is also suggested. First, only stationary kernels have been studied. According to previous studies in non-stationary kernels [41], they can also produce good results in several optimization problems. Second, some aspects of the proposed adaptive kernels can be analyzed more in depth. For example, other weight functions can be applied in *Weighted Best* and *Weighted Mixture*, comparing the actual improvement to the previously suggested utility. Finally, the proposed method can be generalized to other stochastic processes other than GPs.

Acknowledgments

This work has been supported by the Spanish Ministry of Science, Innovation and Universities (project TIN2016-78365-R), and the Basque Government (IT1244-19 and ELKARTEK programs). Jose A. Lozano is also supported by BERC 2018-2021 (Basque government) and BCAM Severo Ochoa accreditation SEV-2017-0718 (Spanish Ministry of Economy and Competitiveness), while Ibai Roman has held a predoctoral grant from the Basque Government.

References

[1] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization*, vol. 13, no. 4, pp. 455–492, Dec. 1998. [Online]. Available: http://link.springer.com/article/10.1023/A%3A1008306431147

[2] J. Mockus, "The Bayesian Approach to Local Optimization," in *Bayesian Approach to Global Optimization*, ser. Mathematics and Its Applications. Springer Netherlands, 1989, no. 37, pp. 125–156. [Online]. Available: http://link.springer.com/chapter/10.1007/978-94-009-0909-0_7

[3] R. Garnett, M. A. Osborne, and S. J. Roberts, "Bayesian Optimization for Sensor Set Selection," in *Proceedings of the 9th ACM/IEEE International*

Conference on Information Processing in Sensor Networks, ser. IPSN '10. New York, NY, USA: ACM, 2010, pp. 209–219. [Online]. Available: http://doi.acm.org/10.1145/1791212.1791238

[4] D. R. Jones, "A Taxonomy of Global Optimization Methods Based on Response Surfaces," *Journal of Global Optimization*, vol. 21, no. 4, pp. 345–383, Dec. 2001. [Online]. Available: http://link.springer.com/article/10.1023/A%3A1012771025575

[5] N. Mahendran, Z. Wang, F. Hamze, and N. D. Freitas, "Adaptive MCMC with Bayesian optimization," in *International Conference on Artificial Intelligence and Statistics*, 2012, pp. 751–760. [Online]. Available: http://machinelearning.wustl.edu/mlpapers/paper_files/AISTATS2012_MahendranWHF12.pdf

[6] E. Meeds and M. Welling, "GPS-ABC: Gaussian Process Surrogate Approximate Bayesian Computation," in *Proceedings of the Thirtieth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-14)*. Corvallis, Oregon: AUAI Press, 2014, pp. 593–602.

[7] C. Audet, J. Denni, D. Moore, A. Booker, and P. Frank, "A surrogate-model-based method for constrained optimization," in *8th Symposium on Multidisciplinary Analysis and Optimization*, ser. Multidisciplinary Analysis Optimization Conferences. American Institute of Aeronautics and Astronautics, 2000. [Online]. Available: http://arc.aiaa.org/doi/abs/10.2514/6.2000-4891

[8] A. Wilson, A. Fern, and P. Tadepalli, "Using Trajectory Data to Improve Bayesian Optimization for Reinforcement Learning," *Journal of Machine Learning Research*, vol. 15, pp. 253–282, 2014. [Online]. Available: http://jmlr.org/papers/v15/wilson14a.html

[9] P. Boyle, "Gaussian Processes for Regression and Optimisation," Ph.D. dissertation, Victoria University of Wellington, 2007. [Online]. Available: https://books.google.es/books?id=Sk6ZMgAACAAJ

[10] R. Garnett, M. A. Osborne, S. Reece, A. Rogers, and S. J. Roberts, "Sequential Bayesian Prediction in the Presence of Changepoints and Faults," *The Computer Journal*, vol. 53, no. 9, pp. 1430–1446, Nov. 2010. [Online]. Available: http://comjnl.oxfordjournals.org/cgi/doi/10.1093/comjnl/bxq003

[11] M. A. Osborne, R. Garnett, and S. J. Roberts, "Gaussian processes for global optimization," in *3rd international conference on learning and intelligent optimization (LION3)*. Citeseer, 2009, pp. 1–15.

[12] C. Durantin, J. Marzat, and M. Balesdent, "Analysis of multi-objective Kriging-based methods for constrained global optimization," *Computational Optimization and Applications*, vol. 63, no. 3, pp. 903–926,

Sep. 2015. [Online]. Available: http://link.springer.com/article/10.1007/s10589-015-9789-6

[13] C. E. Rasmussen and C. K. Williams, *Gaussian processes for machine learning.* MIT Press, 2006.

[14] M. G. Genton, “Classes of Kernels for Machine Learning: A Statistics Perspective,” *J. Mach. Learn. Res.*, vol. 2, pp. 299–312, Mar. 2002. [Online]. Available: http://dl.acm.org/citation.cfm?id=944790.944815

[15] P. Sollich, M. Urry, and C. Coti, “Kernels and learning curves for Gaussian process regression on random graphs,” in *Advances in Neural Information Processing Systems 22*, Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, Eds. Curran Associates, Inc., 2009, pp. 1723–1731. [Online]. Available: http://papers.nips.cc/paper/3840-kernels-and-learning-curves-for-gaussian-process-regression-on-random-graphs.pdf

[16] A. Wilson and R. Adams, “Gaussian Process Kernels for Pattern Discovery and Extrapolation,” in *Proceedings of The 30th International Conference on Machine Learning*, 2013, pp. 1067–1075. [Online]. Available: http://jmlr.org/proceedings/papers/v28/wilson13.html

[17] D. Duvenaud, J. Lloyd, R. Grosse, J. Tenenbaum, and G. Zoubin, “Structure Discovery in Nonparametric Regression through Compositional Kernel Search,” in *Proceedings of The 30th International Conference on Machine Learning*, 2013, pp. 1166–1174. [Online]. Available: http://jmlr.org/proceedings/papers/v28/duvenaud13.html

[18] D. Ginsbourger, C. Helbert, and L. Carraro, “Discrete mixtures of kernels for Kriging-based optimization,” *Qual. Reliab. Engng. Int.*, vol. 24, no. 6, pp. 681–691, 2008. [Online]. Available: http://onlinelibrary.wiley.com/doi/10.1002/qre.945/abstract

[19] M. Hoffman, E. Brochu, and O. D. Freitas, “Portfolio allocation for Bayesian optimization,” in *In UAI*, 2011, pp. 327–336.

[20] B. Shahriari, Z. Wang, M. W. Hoffman, A. Bouchard-Côté, and N. de Freitas, “An Entropy Search Portfolio for Bayesian Optimization,” *arXiv:1406.4625 [cs, stat]*, Jun. 2014, arXiv: 1406.4625. [Online]. Available: http://arxiv.org/abs/1406.4625

[21] F. A. C. Viana, R. T. Haftka, and L. T. Watson, “Efficient global optimization algorithm assisted by multiple surrogate techniques,” *Journal of Global Optimization*, vol. 56, no. 2, pp. 669–689, Mar. 2012. [Online]. Available: http://link.springer.com/article/10.1007/s10898-012-9892-5

[22] E. Brochu, V. M. Cora, and N. de Freitas, “A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application

to Active User Modeling and Hierarchical Reinforcement Learning," *arXiv:1012.2599 [cs]*, Dec. 2010, arXiv: 1012.2599. [Online]. Available: http://arxiv.org/abs/1012.2599

[23] A. Mchutchon and C. E. Rasmussen, "Gaussian Process Training with Input Noise," in *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2011, pp. 1341–1349. [Online]. Available: http://papers.nips.cc/paper/4295-gaussian-process-training-with-input-noise.pdf

[24] H. J. Kushner, "A New Method of Locating the Maximum Point of an Arbitrary Multipeak Curve in the Presence of Noise," *Journal of Basic Engineering*, vol. 86, no. 1, pp. 97–106, Mar. 1964. [Online]. Available: http://dx.doi.org/10.1115/1.3653121

[25] J. Mockus, V. Tiesis, and A. Zilinskas, "The application of Bayesian methods for seeking the extremum," in *Towards Global Optimization*. Elsevier, 1978, vol. 2, pp. 117–129.

[26] N. Srinivas, A. Krause, S. Kakade, and M. Seeger, "Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, 2010, pp. 1015–1022. [Online]. Available: http://www.icml2010.org/papers/422.pdf

[27] ——, "Information-Theoretic Regret Bounds for Gaussian Process Optimization in the Bandit Setting," *IEEE Transactions on Information Theory*, vol. 58, no. 5, pp. 3250–3265, May 2012.

[28] Z. Wang and N. de Freitas, "Theoretical Analysis of Bayesian Optimisation with Unknown Gaussian Process Hyper-Parameters," *arXiv:1406.7758 [cs, stat]*, Jun. 2014, arXiv: 1406.7758. [Online]. Available: http://arxiv.org/abs/1406.7758

[29] I. Murray and R. P. Adams, "Slice sampling covariance hyperparameters of latent Gaussian models," in *Advances in Neural Information Processing Systems 23*, J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, Eds. Curran Associates, Inc., 2010, pp. 1732–1740. [Online]. Available: http://papers.nips.cc/paper/4114-slice-sampling-covariance-hyperparameters-of-latent-gaussian-models.pdf

[30] K. Eggensperger, M. Feurer, F. Hutter, J. Bergstra, J. Snoek, H. Hoos, and K. Leyton-Brown, "Towards an empirical foundation for assessing Bayesian Optimization of hyperparameters," in *NIPS Workshop on Bayesian Optimization in Theory and Practice*, 2013.

[31] J. Bergstra, D. Yamins, and D. D. Cox, “Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms,” in *Proceedings of the 12th Python in Science Conference*, 2013, pp. 13–20.

[32] O. Roustant, D. Ginsbourger, and Y. Deville, “DiceKriging, DiceOptim: two R packages for the analysis of computer experiments by kriging-based metamodelling and optimization,” *Journal of Statistical Software*, vol. 51, no. 1, p. 54p, 2012, http://cran.stat.sfu.ca/web/packages/DiceKriging/DiceKriging.pdf. [Online]. Available: http://hal-emse.ccsd.cnrs.fr/emse-00741762

[33] J. Snoek, H. Larochelle, and R. P. Adams, “Practical Bayesian Optimization of Machine Learning Algorithms,” in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 2951–2959. [Online]. Available: http://papers.nips.cc/paper/4522-practical-bayesian-optimization-of-machine-learning-algorithms.pdf

[34] R. Martinez-Cantin, “BayesOpt: A Bayesian Optimization Library for Nonlinear Optimization, Experimental Design and Bandits,” *Journal of Machine Learning Research*, vol. 15, pp. 3735–3739, 2014. [Online]. Available: http://jmlr.org/papers/v15/martinezcantin14a.html

[35] M. W. Hoffman and B. Shahriari, “Modular mechanisms for Bayesian Optimization,” in *NIPS Workshop on Bayesian Optimization*, 2014.

[36] P. Bratley and B. L. Fox, “Algorithm 659: Implementing Sobol’s Quasirandom Sequence Generator,” *ACM Trans. Math. Softw.*, vol. 14, no. 1, pp. 88–100, Mar. 1988. [Online]. Available: http://doi.acm.org/10.1145/42288.214372

[37] J. Demšar, “Statistical Comparisons of Classifiers over Multiple Data Sets,” *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, 2006. [Online]. Available: http://dl.acm.org/citation.cfm?id=1248547.1248548

[38] W. H. Kruskal and W. A. Wallis, “Use of Ranks in One-Criterion Variance Analysis,” *Journal of the American Statistical Association*, vol. 47, no. 260, pp. 583–621, 1952. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/01621459.1952.10483441

[39] B. Calvo and G. Santafé, “scmamp: Statistical Comparison of Multiple Algorithms in Multiple Problems,” *The R Journal*, vol. 8, no. 1, pp. 248–256, 2016. [Online]. Available: https://journal.r-project.org/archive/2016/RJ-2016-017/index.html

[40] S. García and F. Herrera, “An extension on ”statistical comparisons of classifiers over multiple data sets” for all pairwise comparisons,” *Journal of Machine Learning Research*, vol. 9, no. Dec, pp. 2677–2694, 2008. [Online]. Available: http://digibug.ugr.es/handle/10481/32916

[41] C. J. Paciorek and M. J. Schervish, "Nonstationary Covariance Functions for {G}aussian {P}rocess Regression," in *Advances in Neural Information Processing Systems 16*, S. Thrun, L. K. Saul, and B. Schölkopf, Eds. MIT Press, 2004, pp. 273–280. [Online]. Available: http://papers.nips.cc/paper/2350-nonstationary-covariance-functions-for-gaussian-process-regression.pdf