Bayesian Multi-Scale Optimistic Optimization

Ziyu Wang
University of Oxford

Babak Shakibi
University of British Columbia

Lin Jin
Rocket Gaming Systems

Nando de Freitas
University of Oxford

Abstract

Bayesian optimization is a powerful global optimization technique for expensive black-box functions. One of its shortcomings is that it requires auxiliary optimization of an acquisition function at each iteration. This auxiliary optimization can be costly and very hard to carry out in practice. Moreover, it creates serious theoretical concerns, as most of the convergence results assume that the exact optimum of the acquisition function can be found. In this paper, we introduce a new technique for efficient global optimization that combines Gaussian process confidence bounds and treed simultaneous optimistic optimization to eliminate the need for auxiliary optimization of acquisition functions. The experiments with global optimization benchmarks and a novel application to automatic information extraction demonstrate that the resulting technique is more efficient than the two approaches from which it draws inspiration. Unlike most theoretical analyses of Bayesian optimization with Gaussian processes, our finite-time convergence rate proofs do not require exact optimization of an acquisition function. That is, our approach eliminates the unsatisfactory assumption that a difficult, potentially NP-hard, problem has to be solved in order to obtain vanishing regret rates.

1 Introduction

We consider the problem of approximating the maximizer of a deterministic black-box function $f : \mathcal{X} \mapsto \mathbb{R}$. The function f can be evaluated point-wise, but it is assumed to be expensive to evaluate. More precisely, we assume that we are given a finite budget of n possible function evaluations.

This global optimization problem can be treated within the framework of sequential design. In this context, by allowing $\mathbf{x}_t \in \mathcal{X}$ to depend on previous points and corresponding function evaluations $\mathcal{D}_{t-1} = \{(\mathbf{x}_1, f(\mathbf{x}_1)), \ldots, (\mathbf{x}_{t-1}, f(\mathbf{x}_{t-1}))\}$, the algorithm constructs a sequence $\mathbf{x}_{1:n} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ and returns the element $\mathbf{x}(n)$ of highest possible value. That is, it returns the value $\mathbf{x}(n)$ that minimizes the loss:

$$r_n = \sup_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}) - f(\mathbf{x}(n)).$$

This loss is not the same as the cumulative regret used often in the online learning literature: $R_n = n \sup_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}) - \sum_{t=1}^{n} f(\mathbf{x}(t))$.

Bayesian optimization (BO) is a popular sequential design strategy for global optimization; see Brochu et al. (2009) for an introductory treatment. Since the objective function f is unknown, the Bayesian strategy is to treat it as a random function and place a prior over it. The prior captures our beliefs about the behaviour of the function. After gathering the function evaluations $\mathcal{D}_{t-1}$, the prior is updated to form the posterior distribution over f. The posterior distribution, in turn, is used to construct an *acquisition function* that determines what the next query point $\mathbf{x}_t$ should be. Examples of acquisition functions include probability of improvement, expected improvement, Bayesian expected losses, upper confidence bounds (UCB), and dynamic portfolios of these (Močkus, 1982; Jones, 2001; Garnett et al., 2010; Srinivas et al., 2010; Chen et al., 2012; Hoffman et al., 2011). If we were to implement Thompson sampling strategies (May et al., 2011; Kaufmann et al., 2012; Agrawal & Goyal, 2013) for Gaussian processes (GPs), we would also encounter the difficult problem of having to find the maximizer of a sample from the GP at each iteration, unless we were considering only a finite set of query points (Hoffman et al., 2014).

The maximum of the acquisition function is typically found by resorting to discretisation or by means of an *auxiliary optimizer*. For example, Snoek et al. (2012) use discretisation, Bardenet & Kégl (2010) use adaptive grids, Brochu et al. (2007); Martinez-Cantin et al. (2007) and Mahendran et al. (2012) use the DIRECT algorithm of Jones et al. (1993), Lizotte et al. (2011) use a combination of random discretisation and quasi-Newton hill-climbing,

Appearing in Proceedings of the 17^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2014, Reykjavik, Iceland. JMLR: W&CP volume 33. Copyright 2014 by the authors.

Bergstra et al. (2011) and Wang et al. (2013) use the CMA-ES method of Hansen & Ostermeier (2001), Hutter et al. (2011) apply multi-start local search. (Approaches within the framework of Bayesian nonlinear experimental design, such as (Hennig & Schuler, 2012) for finding maxima and (Kueck et al., 2006, 2009; Hoffman et al., 2009) for learning functions and Markov decision processes, have to rely on expensive approximate inference for computing intractable integrals. An analysis of these approaches is beyond the scope of this paper.)

The auxiliary optimization methodology is problematic for several reasons. First, it is difficult to assess whether the auxiliary optimizer has found the maximum of the acquisition function in practice. This creates important theoretical concerns about the behaviour of BO algorithms because the typical theoretical convergence guarantees are only valid on the assumption that the optimum of the acquisition function can be found exactly; see for example Srinivas et al. (2010); Vazquez & Bect (2010) and Bull (2011). Second, running an auxiliary optimizer at each iteration of the BO algorithm can be unnecessarily costly. For any two consecutive iterations, the acquisition function may not change drastically. This questions the necessity of re-starting the auxiliary optimization at each iteration.

Recent *optimistic optimization* methods provide a viable alternative to BO (Kocsis & Szepesvári, 2006; Bubeck et al., 2011; Munos, 2011). Instead of estimating a posterior distribution over the unknown objective function, these methods build space partitioning trees by expanding leaves with high function values or upper-bounds. The term optimistic, in this context, is used to refer to the fact that the algorithms expand at each round leaves that may contain the optimum. Remarkably, a variant of these methods, *Simultaneous Optimistic Optimization* (SOO) by Munos (2011), is able to optimize an objective function globally without knowledge of the function's smoothness. SOO is optimistic at all scales in the sense that it expands several leaves simultaneously, with at most one leaf per level. For this reason, instead of adopting the term "Simultaneous OO" we opt for the descriptive term "Multi-Scale OO".

We will describe SOO in more detail in Section 3. We also note that a stochastic variant of SOO has been recently proposed by Valko et al. (2013), but we restrict the focus of this paper to the deterministic case.

These optimistic optimization methods do not require the auxiliary optimization of acquisition functions. However, due to the lack of a posterior that interpolates between the sampled points, it is conceivable that these methods may not be as competitive as BO in practical domains where prior knowledge is available. This claim does not seem to have been backed up by empirical evidence in the past.

This paper introduces a new algorithm, BaMSOO, which combines elements of BO and SOO. Importantly, it eliminates the need for auxiliary optimization of the acquisition function in BO. We derive theoretical guarantees for the method that do not depend on the assumption that the acquisition function needs to be optimized exactly. The method uses SOO to optimize the objective function directly, but eliminates the need for SOO to sample points that are deemed unfit by Gaussian process posterior bounds. That is, BaMSOO uses the posterior distribution to reduce the number of function evaluations in SOO, thus increasing the efficiency of SOO substantially.

The experiments with benchmarks from the global optimization literature demonstrate that BaMSOO outperforms both GP-UCB and SOO. The paper also introduces a novel application in the domain of knowledge discovery and information extraction. Finally, our theoretical results show that BaMSOO can attain, up to log factors, a polynomial finite sample convergence rate.

2 BO with GP confidence bounds

Classical BO approaches have two ingredients that need to be specified: The prior and the acquisition function. In this work, as in most other works, we adopt Gaussian process (GP) priors. We review GPs very briefly and refer the interested reader to the book of Rasmussen & Williams (2006) for an in-depth treatment. A GP is a distribution over functions specified by its mean function $m(\cdot)$ and covariance $\kappa(\cdot,\cdot)$. More specifically, given a set of points $\mathbf{x}_{1:t}$, with $\mathbf{x}_i \in \mathcal{X} \subseteq \mathbb{R}^D$, we have

$$\mathbf{f}(\mathbf{x}_{1:t}) \sim \mathcal{N}(\mathbf{m}(\mathbf{x}_{1:t}), \mathbf{K}),$$

where $\mathbf{K}$, with entries $\mathbf{K}_{i,j} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$, is the covariance matrix. A common choice of κ in the BO literature is the anisotropic kernel with a vector of known hyper-parameters

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) \quad = \quad \widetilde{\kappa}\left(-(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{D} (\mathbf{x}_i - \mathbf{x}_j)\right), \quad (1)$$

where $\widetilde{\kappa}$ is an isotropic kernel and $\mathbf{D}$ is a diagonal matrix with positive hyper-parameters along the diagonal and zeros elsewhere. Our results apply to squared exponential kernels and Matérn kernels with parameter $\nu \geq 2$. In this paper, we assume that the hyper-parameters are fixed and known in advance. We refer the reader to Martinez-Cantin et al. (2007); Brochu et al. (2010); Wang et al. (2013); Snoek et al. (2012) for different practical approaches to estimate the hyper-parameters.

An advantage of using GPs lies in their analytical tractability. In particular, given observations $\mathcal{D}_t = \{\mathbf{x}_{1:t}, \mathbf{f}_{1:t}\}$, where $f_i = f(\mathbf{x}_i)$, and a new point $\mathbf{x}_{t+1}$, the joint distribution is given by:

$$\begin{bmatrix} \mathbf{f}_{1:t} \\ f_{t+1} \end{bmatrix} \sim \mathcal{N}\left(\mathbf{m}(\mathbf{x}_{1:t+1}), \begin{bmatrix} \mathbf{K} & \mathbf{k} \\ \mathbf{k}^T & \kappa(\mathbf{x}_{t+1}, \mathbf{x}_{t+1}) \end{bmatrix}\right)$$

where $\mathbf{k}^T = [\kappa(\mathbf{x}_{t+1}, \mathbf{x}_1) \cdots \kappa(\mathbf{x}_{t+1}, \mathbf{x}_t)]$. For simplicity, we assume that $\mathbf{m}(\cdot) = \mathbf{0}$. Using the Sherman-Morrison-

Woodbury formula, one can easily arrive at the posterior predictive distribution:

$$f_{t+1}|\mathcal{D}_t, \mathbf{x}_{t+1} \sim \mathcal{N}(\mu(\mathbf{x}_{t+1}|\mathcal{D}_t), \sigma^2(\mathbf{x}_{t+1}|\mathcal{D}_t)),$$

with mean $\mu(\mathbf{x}_{t+1}|\mathcal{D}_t) = \mathbf{k}^T\mathbf{K}^{-1}\mathbf{f}_{1:t}$ and variance $\sigma^2(\mathbf{x}_{t+1}|\mathcal{D}_t) = \kappa(\mathbf{x}_{t+1}, \mathbf{x}_{t+1}) - \mathbf{k}^T\mathbf{K}^{-1}\mathbf{k}$. We can compute the posterior predictive mean $\mu(\cdot)$ and variance $\sigma^2(\cdot)$ exactly for any point $\mathbf{x}_{t+1}$.

At each iteration of BO, one has to re-compute the predictive mean and variance. These two quantities are used to construct the second ingredient of BO: The acquisition function (or utility function). In this work, we report results for the GP-UCB acquisition function $\mathcal{U}(\mathbf{x}|\mathcal{D}_t) = \mu(\mathbf{x}|\mathcal{D}_t) + \sqrt{B_t}\sigma(\mathbf{x}|\mathcal{D}_t)$, which is the upper confidence bound (UCB) on the objective function (Srinivas et al., 2010; de Freitas et al., 2012). We also make use of the lower confidence bound (LCB) which is defined as $\mathcal{L}(\mathbf{x}|\mathcal{D}_t) = \mu(\mathbf{x}|\mathcal{D}_t) - \sqrt{B_t}\sigma(\mathbf{x}|\mathcal{D}_t)$. In these definitions, B_t is such that $f(\mathbf{x})$ is bounded above and below by $\mathcal{U}(\mathbf{x}|\mathcal{D}_t)$ and $\mathcal{L}(\mathbf{x}|\mathcal{D}_t)$ with high probability (de Freitas et al., 2012).

BO selects the next query point by optimizing the acquisition function $\mathcal{U}(\mathbf{x}|\mathcal{D}_t)$. Note that our choice of utility favours the selection of points with high variance (points in regions not well explored) and points with high mean value (points worth exploiting). As mentioned in the introduction, the optimization of the closed-form acquisition function is often carried out by off-the-shelf global optimization procedures, such as DIRECT and CMA-ES.

Many other acquisition functions have been proposed, but they often yield similar results; see for example the works of Močkus (1982) and Jones (2001). The idea of learning portfolios of acquisition functions online was explored by Hoffman et al. (2011). We do not consider these acquisition functions for brevity. The BO procedure is summarized in Algorithm 1.

Algorithm 1 GP-UCB

```
for $t = 1, 2, \ldots$ do
    $\mathbf{x}_{t+1} = \arg\max_{\mathbf{x}\in\mathcal{X}} \mathcal{U}(\mathbf{x}|\mathcal{D}_t)$.
    Augment the data $\mathcal{D}_{t+1} = \{\mathcal{D}_t, (\mathbf{x}_{t+1}, f(\mathbf{x}_{t+1}))\}$
end for
```

Finite sample bounds for GP-UCB were derived by Srinivas et al. (2010). However, the bounds depend on the algorithm being able to optimize the UCB acquisition function, at each iteration, exactly. Unless the action set is discrete, it is unlikely that we will be able to find the global optimum of the UCB with a fixed budget optimization method. That is, we may not be able to guarantee that we can find the exact optimum of the UCB, and hence the theoretical bounds seem to make a very strong assumption in this regard.

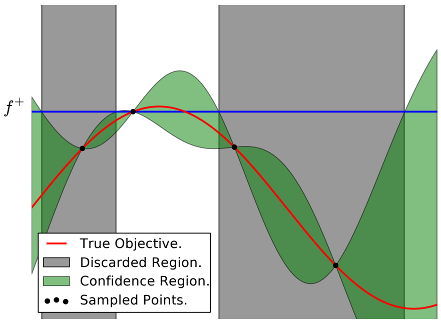

Figure 1: The global shrinking method of de Freitas et al. (2012). If the unknown objective function lies within the (green) confidence bounds with high probability, we can discard regions of the space where the upper bound is lower than the best lower bound encountered thus far.

2.1 Shrinking feasible regions

de Freitas et al. (2012) introduced a different GP-based scheme to trade off exploration and exploitation. Instead of optimizing the acquisition function, they proposed to sample the objective function using a finite lattice within a feasible region R. The feasible region at the t^{th} iteration is defined as

$$R_t = \{\mathbf{x} : \mu_t(\mathbf{x}) + B_t\sigma_t(\mathbf{x}) > \sup_{\mathbf{x}\in R_{t-1}} \mu_t(\mathbf{x}) - B_t\sigma_t(\mathbf{x})\}.$$

That is, one should only search in the region where the upper bound is greater than the best lower bound encountered thus far, as illustrated in Figure 1. With high probability, the optimizer lies within R_t.

de Freitas et al. (2012) proved that if we double the density of points in the lattice at each iteration, the feasible region shrinks very quickly. More precisely, they showed that the simple regret vanishes at an exponential rate and that the cumulative regret is bounded by a constant.

With this approach, they did not have to resort to optimizing an acquisition function. However, even in moderate dimensions, their algorithm is impractical since the lattice often becomes too large to be sampled in a reasonable amount of time.

In this paper, we will argue that to overcome this problem, an optimistic strategy may have to be employed. Such a strategy enables us to sample the most promising regions first, so as to avoid the computational cost associated with covering the whole space. In the next section, we begin our discussion of optimistic strategies.

3 Simultaneous optimistic optimization

Deterministic optimistic optimization (DOO) and simultaneous optimistic optimization (SOO) are tree-based space

Algorithm 2 SOO

```
Evaluate $f(\mathbf{x}_{0,0})$
Initialize the tree $\mathcal{T}_1 = \{0, 0\}$
Set $n = 1$
while true do
    Set $\nu_{\max} = -\infty$
    for $h = 0 : \min\{\text{depth}(\mathcal{T}_n), h_{\max}(n)\}$ do
        Select $(h, j) = \arg\max_{j \in \{j | (h,j) \in L_n\}} f(\mathbf{x}_{h,j})$
        if $f(\mathbf{x}_{h,j}) > \nu_{\max}$ then
            Evaluate the children of $(h, j)$
            Add the children of $(h, j)$ to $\mathcal{T}_n$
            Set $\nu_{\max} = f(\mathbf{x}_{h,j})$
            Set $n = n + 1$
        end if
    end for
end while
```

partitioning methods for black-box function optimization (Munos, 2011, 2014). They were inspired by the UCT algorithm, which enjoyed great success in planning (Kocsis & Szepesvári, 2006). UCT was shown to have no finite-time guarantees by Coquelin & Munos (2007). This prompted the development of a range of optimistic, in the face of uncertainty, approaches. The term optimism, here, refers to the fact that the strategies expand at each round tree cells that may contain the optimum.

DOO and SOO partition the space $\mathcal{X}$ hierarchically by building a tree. Let us assume that each node of the tree has k children. A node (h, j) at level h of the tree has children $\{(h+1, kj+i)\}_{0 \le i < k-1}$. The children partition the parent cell $X_{h,j}$ into cells $\{X_{h+1,kj+i},\ 0 \le i < k-1\}$. The root cell is the entire space $\mathcal{X}$. A node is always evaluated at the center of the cell, which we denote as $\mathbf{x}_{h,j}$.

Instead of assuming that the target function is a sample from a GP, DOO and SOO assume the existence of a symmetric semi-metric ℓ such that $f(\mathbf{x}^*) - f(\mathbf{x}) \le \ell(\mathbf{x}, \mathbf{x}^*)$ where $\mathbf{x}^*$ is the maximizer of f. Although, SOO assumes that ℓ exists, it does not require explicit knowledge of it.

DOO on the other hand does require knowledge of ℓ. DOO builds a tree $\mathcal{T}_n$ incrementally, where n denotes the index over node expansions. DOO expands a leaf (h, j) from the set of leaves L_n (nodes whose children are not in $\mathcal{T}_n$) if it has the the highest upper bound: $f(\mathbf{x}_{h,j}) + \sup_{\mathbf{x} \in X_{h,j}} \ell(\mathbf{x}_{h,j}, \mathbf{x})$. This value for any cell containing $\mathbf{x}^*$ upper bounds the best function value f^*. The performance of DOO depends crucially on our knowledge of the true local smoothness of f. SOO aims to overcome the difficulty of having to know the true local smoothness.

SOO, as summarized in Algorithm 2, expands several leaves simultaneously. When a node is expanded, its children are evaluated. At each round, SOO expands at most one leaf per level, and a leaf is expanded only if it has the largest value among all leaves of the same or lower depths. The SOO algorithm takes as input a function $n \to h_{\max}(n)$, which limits the maximum height of the tree after n node expansions. $h_{\max}(n)$ defines a tradeoff between deep versus broad exploration. At the end of the finite horizon, SOO returns the $\mathbf{x}$ with the highest objective function value. Figure 2 illustrates the application of SOO to a simple 1-dimensional optimization problem.

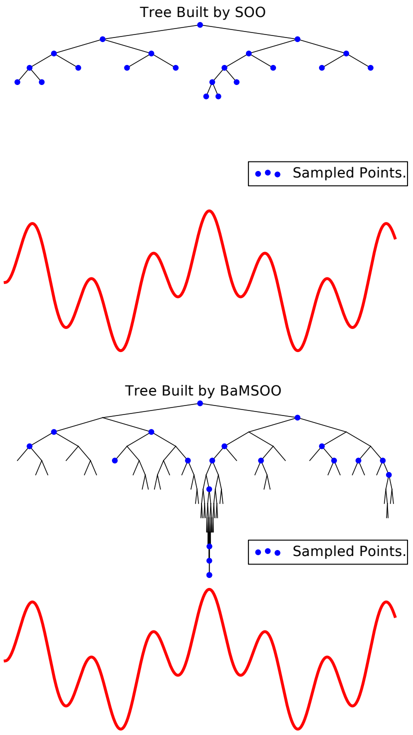

Figure 2: [TOP]: The tree built by SOO when optimizing the function $f(x) = \frac{1}{2}\sin(15x)\sin(27x)$ in $[0, 1]$. [BOTTOM]: The tree built by BaMSOO. The 20 blue dots represent nodes where the objective was evaluated. BaMSOO, in comparison, does not evaluate the objective function for points known to be sub-optimal with high probability. Hence, BaMSOO can achieve a better coverage of the search space with the same number of function evaluations as SOO.

4 BaMSOO

SOO offers a different way of trading off exploration and exploitation that does not require the optimization of an acquisition function. However, it does not utilize all the information brought in by the previously evaluated points effectively. To improve upon SOO in practice, we consider the additional assumption that the objective function is a sample from a GP prior.

We define the LCB and UCB to be $\mathcal{L}_N(\mathbf{x}|\mathcal{D}_t) = \mu(\mathbf{x}|\mathcal{D}_t) - B_N\sigma(\mathbf{x}|\mathcal{D}_t)$ and $\mathcal{U}_N(\mathbf{x}|\mathcal{D}_t) = \mu(\mathbf{x}|\mathcal{D}_t) + B_N\sigma(\mathbf{x}|\mathcal{D}_t)$ where $B_N = \sqrt{2\log(\pi^2 N^2/6\eta)}$ and $\eta \in (0, 1)$.

The BaMSOO algorithm is very similar to SOO. As with SOO, we only evaluate the cell at the center point. However, when a node's UCB is less than the function value of the best point already sampled, denoted f^+, we do not evaluate the objective function at this node because with high probability the center point is sub-optimal. Instead, we simply assign to this node its LCB value. Note that if the center-point of a cell is sub-optimal, the cell may still contain the optimizer. Hence this cell must also be further expanded in subsequent iterations. To manage these two types of node in the pseudo-code (see Algorithm 3), we introduce a place-holder function g which is set to f when the UCB of the cell of interest is bigger than f^+, and it is set to the LCB of the node otherwise. For clarity, we remind the reader that the indices N, k, t and n are over node evaluations, branches (children), function evaluations and node expansions respectively.

In the pseudocode, we have highlighted in blue the additional lines of code brought in by BaMSOO. Effectively, BamSOO only involves a slight modification of SOO (Algorithm 2) provided we have GP routines to evaluate the LCB and UCB.

We found the assignment of the LCB values to nodes that do worse than f^+ to work well in practice. For this reason our presentation, experiments and theory focus on this choice.

BaMSOO improves upon SOO by making use of the available information more efficiently. Moreover, by using an optimistic proposal, it avoids the need to sample exhaustively before shrinking the feasible region as in (de Freitas et al., 2012). Figure 2 illustrates how BaMSOO can cover the search space more effectively, even though it incurs the same number of expensive function evaluations as SOO.

Algorithm 3 BaMSOO

```
Set $g_{0,0} = f(\mathbf{x}_{0,0})$
Set $f^+ = g_{0,0}$
Initialize the tree $\mathcal{T}_1 = \{0, 0\}$
Set $t = 1, n = 1, N = 1$, and $\mathcal{D}_t = \{(\mathbf{x}_{0,0}, g(\mathbf{x}_{0,0}))\}$
while true do
  Set $\nu_{\max} = -\infty$.
  for $h = 0$ to $\min\{\text{depth}(\mathcal{T}_n), h_{\max}(n)\}$ do
    Select $(h, j) = \arg\max_{j \in \{j|(h,j) \in L_n\}} g(\mathbf{x}_{h,j})$
    if $g(x_{h,j}) > \nu_{\max}$ then
      for $i = 0$ to $k - 1$ do
        Set $N = N + 1$
        if $\mathcal{U}_N(\mathbf{x}_{h+1,kj+i}|\mathcal{D}_t) \geq f^+$ then
          Set $g(\mathbf{x}_{h+1,kj+i}) = f(\mathbf{x}_{h+1,kj+i})$
          Set $t = t + 1$
          $\mathcal{D}_t = \{\mathcal{D}_{t-1}, (\mathbf{x}_{h+1,kj+i}, g(\mathbf{x}_{h+1,kj+i}))\}$
        else
          Set $g(\mathbf{x}_{h+1,kj+i}) = \mathcal{L}_N(\mathbf{x}_{h+1,kj+i}|\mathcal{D}_t)$
        end if
        if $g(\mathbf{x}_{h+1,kj+i}) > f^+$ then
          Set $f^+ = g(\mathbf{x}_{h+1,kj+i})$
        end if
      end for
      Add the children of $(h, j)$ to $\mathcal{T}_n$
      Set $\nu_{\max} = g(\mathbf{x}_{h,j})$
      Set $n = n + 1$
    end if
  end for
end while
```

5 Analysis

In this section, we provide an overview of the theoretical analysis of BaMSOO, which appears in the Appendix. Our discussion here will focus on our assumptions. At the end of this section, we will present the main result and sketch the proof coarsely.

We denote the global maximum by $f^* = \sup_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})$ and the maximizer by $\mathbf{x}^* = \arg\max_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})$.

We make similar assumptions to those made by de Freitas et al. (2012). As in their case, we make the global assumption that the objective function is a sample from a GP and a local assumption about the behavior of the objective near the optimum.

Assumption 1 (Conditions on the GP kernel)**.** *$\mathcal{X} \subseteq \mathbb{R}^D$ is a compact set, and κ is a kernel on $\mathbb{R}^D$ that is twice differentiable along the diagonal such that $\partial_{\mathbf{x}}\partial_{\mathbf{x}'}\kappa(\mathbf{x}, \mathbf{x}')|_{\mathbf{x}=\mathbf{x}'}$ exists.*

Assumption 2 (Local smoothness of f)**.** *$f \sim GP(0, \kappa)$ is a continuous sample on $\mathcal{X}$ that has a unique global maximum $\mathbf{x}^*$, such that $f^* - c_1\|\mathbf{x} - \mathbf{x}^*\|_2^\alpha \leq f(\mathbf{x})\ \forall \mathbf{x} \in \mathcal{X}$ and $f(\mathbf{x}) \leq f^* - c_2\|\mathbf{x} - \mathbf{x}^*\|_2^2\ \forall \mathbf{x} \in \mathcal{B}(\mathbf{x}^*, \rho)$ for some constants c_1, c_2, $\rho > 0$ and $\alpha \in \{1, 2\}$. Also $f^* - \max_{\mathbf{x} \in \mathcal{X} \setminus \mathcal{B}(\mathbf{x}^*, \rho)} f(\mathbf{x}) > \epsilon_0$ for some $\epsilon_0 > 0$.*

As argued by de Freitas et al. (2012), in many practical cases the local conditions follow almost surely from the global condition. For example, if we were to employ the Matern kernel with $\nu > 2$ or a kernel that is 6 times differentiable along the diagonal, we would have that the samples of the GPs are twice differentiable with probability one. The first case was shown by (Adler & Taylor, 2007, Theorem 1.4.2) and (Stein, 1999, §2.6), while the second result was shown by (Ghosal & Roy, 2006, Theorem 5). If the $\mathbf{x}^*$ lies in the interior of $\mathcal{X}$, then the Hessian of f at $\mathbf{x}^*$ would be almost surely non-singular as at least one of the eigenvalues of the Hessian is a co-dimension 1 condition in the space of all functions that are smooth at a given point (de Freitas et al., 2012). In this case, we would have that

$$f^* - c_1\|\mathbf{x} - \mathbf{x}^*\|_2^\alpha \leq f(\mathbf{x}) \leq f^* - c_2\|\mathbf{x} - \mathbf{x}^*\|_2^2$$

with $\alpha = 2$.

If $\mathbf{x}^*$ lies on the boundary of $\mathcal{X}$ which we assume to be smooth, then $\nabla f(\mathbf{x}^*) \neq 0$ since the additional event of the vanishing of $\nabla f(\mathbf{x}^*)$ is a co-dimension d phenomenon in the space of functions with global maximum at $\mathbf{x}^*$ (de

Freitas et al., 2012). In this case, we would have that

$$f^* - c_1\|\mathbf{x} - \mathbf{x}^*\|_2^\alpha \le f(\mathbf{x}) \le f^* - c_2\|\mathbf{x} - \mathbf{x}^*\|_2^2$$

with $\alpha = 1$.

Finally, a sample from a GP on a compact domain has a unique maximum with probability one. This is because the space of continuous functions on a compact domain that attain their global maximum at more than one point have co-dimension 1 in the space of all continuous functions on that domain (de Freitas et al., 2012).

The subsequent assumptions are about the hierarchical partitioning of the search space. They are the same as Assumptions 3 and 4 in Munos (2011).

Assumption 3 (Bounded diameters)**.** *There exists a decreasing sequence $\delta(h) > 0$, such that for any depth $h \ge 0$, for any cell $X_{h,i}$ of depth h, we have $\sup_{\mathbf{x}\in X_{h,i}} \ell(\mathbf{x}_{h,i}, \mathbf{x}) \le \delta(h)$. Here $\ell(\mathbf{x}, \mathbf{y}) := c_1\|\mathbf{x} - \mathbf{y}\|_2^\alpha$ where $\alpha \in \{1, 2\}$ and $\delta(h) = c\gamma^h$ for some constant $c > 0$ and $\gamma \in (0, 1)$.*

Assumption 4 (Well-shaped cells)**.** *There exists $\nu > 0$ such that for any depth $h \ge 0$, any cell $X_{h,i}$ contains an ℓ-ball of radius $\nu\delta(h)$ centered in $X_{h,i}$.*

Note that depending on the value of α, γ would have to take on a different value for Assumptions 3 and 4 to be satisfied. Regardless of the choice of α and as illustrated in Example 1 of Bubeck et al. (2011), Assumptions 3 and 4 are easy to satisfy in practice; for example when $\mathbf{x} \in [0, 1]^D$ and the split is done along the largest dimension of a cell. This is the case in all our experiments.

Assumption 2 together with Assumptions 3 and 4 impose a "*near optimality*" condition as defined by Munos (2011).

We can now present our main result, which is in the form of a corollary to Theorem 1 in the Appendix.

Corollary 1. *Let $d = -(D/4 - D/\alpha)$ and $h_{\max}(n) = n^\epsilon$. Given Assumptions $1 - 4$, we have that with probability at least $1 - \eta$, the loss of BaMSOO is $\mathcal{O}\left(n^{-\frac{1-\epsilon}{d}} \log^{\frac{\alpha}{4-\alpha}}(n^2/\eta)\right)$.*

It is worth pointing out that the result presented in Corollary 1 is based on the number of node expansions n instead of the number of function evaluations. The theory can therefore be strengthened.

If $\alpha = 2$ and $\epsilon = 1/2$, then the above result translates to $\mathcal{O}\left(n^{-\frac{2}{D}} \log(n^2/\eta)\right)$. If $\alpha = 1$ with ϵ being the same as before, then the rate of convergence becomes $\mathcal{O}\left(n^{-\frac{2}{3D}} \log^{\frac{1}{3}}(n^2/\eta)\right)$.

The structure of the proof follows that in Munos (2011). Let $\mathbf{x}_h^*$ denote the optimal node at level h (that is, the node at height h in the branch that contains the optimum $\mathbf{x}^*$). Our proof shows that once $\mathbf{x}_h^*$ is expanded, it does not take long for $\mathbf{x}_{h+1}^*$ to be expanded. Once an optimal node $\mathbf{x}_h^*$ is expanded, by Assumptions 2 and 3, we have that the loss of BaMSOO is no worse than $\delta(h)$.

The main difficulty of the proof lies in the fact that we sometimes do not sample nodes when their UCB values are less than the best observed value. In this case, we can no longer make the claim that an optimal node is expanded soon after its parent. This is because when a node is not expanded, its LCB can be very low due to a high standard deviation. Fortunately, we can show that this is not the case for optimal nodes in the optimal region. This is accomplished by showing that the standard deviation at a point is no more than its distance to the nearest sampled point up to a constant factor (shown in Lemma 3). This enables us to show that every optimal node in the optimal region must have a low standard deviation. Given this result, we can adopt the proof structure outlined in Munos (2011).

6 Experiments with global optimization benchmarks

In this section, we validate the proposed algorithm with a series of experiments that compare the three algorithms (GP-UCB, SOO, BaMSOO) on global optimization benchmarks. We have omitted the feasible region shrinking algorithm (described in Section 2.1) as it is not practical for problems of even moderate dimensions. We have also omitted comparisons to PI and EI as these appear in Hoffman et al. (2011) for the optimization benchmarks described in this paper.

In our experiments, we used the same hyper-parameters in GP-UCB and BaMSOO for each test function. We also randomized the initial sample point for BaMSOO and GP-UCB so that they are not deterministic. To optimize the acquisition function for GP-UCB, we used DIRECT followed by a local optimization method using gradients.

We use 5 test functions: Branin, Rosenbrock, Hartmann3, Hartmann6, and Shekel. All of these test functions are common in the global optimization literature and with the exception of the Rosenbrock, they are all multi-modal. [1]

We rescaled the domain of each function to the $[0, 1]^D$ hypercube, and we used the log distance to the true optimum as our evaluation metric. This metric is defined as $\log_{10}(f^* - f^+)$ where f^+ is the best objective value sampled so far and f^* is the true maximum value of the objective. For each test function, we repeat our experiments 50 times for GP-UCB and BaMSOO and run SOO once as SOO is a deterministic strategy. We plot the mean and a confidence bound of one standard deviation of our metric

[1] Detailed information about the test functions is available at the following website: `http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO_files/Page364.htm`.

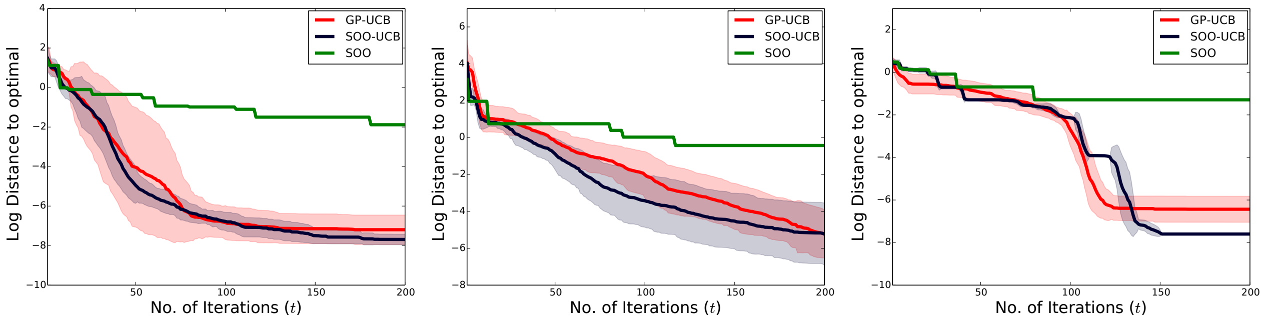

Figure 3: Comparison of GP-UCB, SOO, and BaMSOO on multi-modal test functions of low dimensionality (Branin, Rosenbrock and Hartmann3D). GP-UCB and BaMSOO perform similarly whereas SOO does poorly. The poor performance of SOO is caused by having weaker assumptions on the smoothness of the objective function. The good performance of GP-UCB indicates that when the dimensionality is low optimizing the acquisition function is reasonable.

across all the runs for all the tests.

For simplicity, we only consider binary-trees for space partitioning in SOO and BaMSOO. Specifically, the largest dimension in the parent's cell is split to create two children.

First, we test the global optimization schemes on 3 test functions with low dimensionality: Branin, Rosenbrock and Hartmann3. The Branin function (Jones, 2001) is a common benchmark for Bayesian optimization and has 2 dimensions. The Rosenbrock function is a commonly used non-convex test function for local optimization algorithms, and although it is unimodal, its optimum lies in a long narrow valley, which makes the function hard to optimize. Finally, the Hartmann3 function is 3-dimensional and has four local optima.

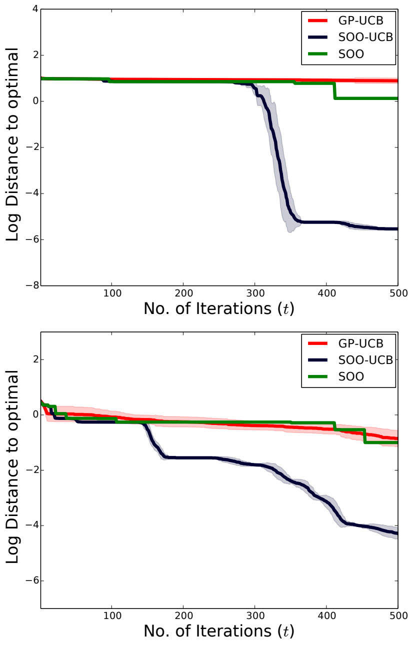

Figure 4: Comparison of GP-UCB, SOO, and BaMSOO on multi-modal test functions of moderate dimensionality: 4D Shekel function (top) and 6D Hartmann function (bottom). Here, GP-UCB performs poorly. This is due in part to the hardness of optimizing the acquisition function.

As we can see from Figure 3, BaMSOO performs competitively against GP-UCB on these low dimensional test functions. Both BaMSOO and GP-UCB achieve very high accuracies of up to 10^{-8} in terms of the distance to the optimal objective value. In comparison, SOO, due to the lack of a strong prior assumption, cannot take advantage of the points sampled and thus is lagging behind.

In the experiments shown in Figure 4, we compare the approaches in consideration on the Shekel function and the Hartmann6 function. The Shekel function is 4-dimensional and has 10 local optima. The Hartmann6 function is 6-dimensional, as the name suggests, and has 6 local optima. On these higher dimensional problems, the performance of GP-UCB begins to dwindle. Despite the increase in dimensionality, BaMSOO is still able to optimize the test functions to a relatively high precision. SOO does not perform as well as BaMSOO again because of its weak assumptions. The poor performance of GP-UCB on these two test functions may be due in part to the inability of a global optimizer to optimize the acquisition function exactly in each iteration. As the dimensionality increases, so is the difficulty of optimizing a non-convex function globally as the cost of covering the space grows exponentially. The optimization of the acquisition function through algorithms like DIRECT demands the repartitioning of the space in each iteration. To reach a finer granularity, we either have to sacrifice speed by building very fine partitions in each iteration or accuracy by using coarser partitions.

The proposed approach is not only competitive with GP-UCB in terms of effectiveness, it is also more computationally efficient. As we can see in Table 1, BaMSOO is about 10-40 times faster than GP-UCB on the test func-

Table 1: Time required for the test functions measured in seconds. SOO is very fast as it does not maintain a GP. BaMSOO maintains a GP to produce more accurate posterior estimates and is hence slower. The rejection of proposals also results in bigger trees, further slowing down the algorithm. GP-UCB is slow compared to the other two algorithms as it not only maintains a GP but also optimizes its acquisition function at each iteration.

Algorithm	Branin	Rosenbrock	Hartmann3	Hartmann6	Shekel
GP-UCB	29.9438	29.5716	34.0311	115.2402	100.7770
BaMSOO	3.0680	3.4693	3.9722	2.0918	3.8951
SOO	0.1810	0.1835	0.1871	0.4313	0.4350

tions that we have experimented with. This is because instead of optimizing the acquisition function in each iteration the SOO algorithm, that sits inside, only optimizes once. BaMSOO, however, is much slower than SOO. This is because BaMSOO also employs a GP to reject points proposed by SOO. To sample one point, SOO may have to propose many points before one is accepted. For this reason, BaMSOO would build much bigger trees compared to SOO and it is therefore slower.

7 Application to term extraction

In this section, we evaluate the performance of the BaMSOO algorithm on optimizing the parameters in a term extraction algorithm. Term extraction is the process of analyzing a text corpus to find terms, where terms correspond to cohesive sequences of words describing entities of interest. Term extraction tools are widely used in industrial text mining and play a fundamental role in the construction of knowledge graphs and semantic search products. Recently Parameswaran et al. (2010) proposed a term extraction method, and showed that it outperforms state-of-the-art competitors, but their method has many free parameters that require manual adjustment. Here, we compare the performance of BaMSOO, GP-UCB and SOO in automatically tuning the 4 primary free parameters of the algorithm (support-thresholds). We define our deterministic objective function to be the F-score of the extracted terms, which is a weighted average of precision and recall. Precision is calculated using a predefined set of correct terms and recall is estimated by simply normalizing the number of extracted correct terms to be in the range [0,1]. We run the experiment on the GENIA corpus (Kim et al., 2003), which is a collection of 2000 abstracts from biomedical articles. The results of the experiment are shown in Figure 5. It is evident from this figure that BaMSOO outperforms GP-UCB and SOO in this application.

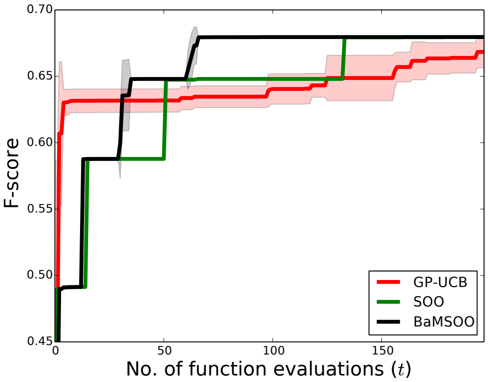

Figure 5: Comparison of GP-UCB, SOO, and BaMSOO on optimizing 4 parameters in term extraction from the GENIA corpus using a term extraction algorithm by (Parameswaran et al., 2010). In this plot, higher is better.

8 Discussion

This paper introduced a new global optimization algorithm BaMSOO, which does not require the auxiliary optimization of either acquisition functions or samples from the GP. In trials with benchmark functions from the global optimization literature, the new algorithm outperforms standard BO with GPs and SOO, while being computationally efficient. The paper also provided a theoretical analysis proving that the loss of BaMSOO decreases polynomially.

The careful reader may have noticed that, despite the effectiveness of BaMSOO in the experiments, the convergence rate of BaMSOO is not as good as that of SOO for $\alpha = 2$. This is because we were only able to prove that the standard deviation at a point decreases linearly, instead of quadratically, when a nearby point is sampled (Lemma 3 in the Appendix). Since by assumption the objective function behaves quadratically in the optimal region, the linear decrease of the standard deviation gives rise to a suboptimal convergence rate. It is also interesting to note that the same type of bound on the standard deviation was used by Bull (2011), who achieved similar convergence rates to the ones in this paper. de Freitas et al. (2012) showed that if the samples form a δ-cover on a subset of $\mathcal{D} \subseteq \mathcal{X}$, then the standard deviation of all points on $\mathcal{D}$ is bounded by a quadratic term $\frac{Q}{4}\delta^2$. Via this observation, the authors achieved a geometric convergence rate. The requirement of the δ-cover, however, renders their algorithm impractical. Finding a *practical* GP-based algorithm that achieves geometric convergence rates remains an open problem.

References

Adler, R. J. and Taylor, J. E. *Random Fields and Geometry*. Springer, 2007.

Agrawal, S. and Goyal, N. Thompson sampling for contextual bandits with linear payoffs. In *International Conference on Machine Learning*, 2013.

Bardenet, R. and Kégl, B. Surrogating the surrogate: accelerating Gaussian-process-based global optimization with a mixture cross-entropy algorithm. In *International Conference on Machine Learning*, pp. 55–62, 2010.

Bergstra, J., Bardenet, R., Bengio, Y., and Kégl, B. Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems*, pp. 2546–2554, 2011.

Brochu, E., de Freitas, N., and Ghosh, A. Active preference learning with discrete choice data. In *Advances in Neural Information Processing Systems*, pp. 409–416, 2007.

Brochu, E., Cora, V. M., and de Freitas, N. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. Technical Report UBC TR-2009-23 and arXiv:1012.2599v1, Dept. of Computer Science, University of British Columbia, 2009.

Brochu, E., Brochu, T., and de Freitas, N. A Bayesian interactive optimization approach to procedural animation design. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 103–112, 2010.

Bubeck, S., Munos, R., Stoltz, G., and Szepesvari, C. X-armed bandits. *Journal of Machine Learning Research*, 12:1655–1695, 2011.

Bull, A. D. Convergence rates of efficient global optimization algorithms. *Journal of Machine Learning Research*, 12:2879–2904, 2011.

Chen, B., Castro, R.M., and Krause, A. Joint optimization and variable selection of high-dimensional Gaussian processes. In *International Conference on Machine Learning*, 2012.

Coquelin, P.A. and Munos, R. Bandit algorithms for tree search. In *Uncertainty in Artificial Intelligence*, pp. 67–74, 2007.

de Freitas, N., Smola, A., and Zoghi, M. Exponential regret bounds for Gaussian process bandits with deterministic observations. In *International Conference on Machine Learning*, 2012.

Garnett, R., Osborne, M. A., and Roberts, S. J. Bayesian optimization for sensor set selection. In *ACM/IEEE International Conference on Information Processing in Sensor Networks*, pp. 209–219. ACM, 2010.

Ghosal, S. and Roy, A. Posterior consistency of Gaussian process prior for nonparametric binary regression. *The Annals of Statistics*, 34:2413–2429, 2006.

Hansen, N. and Ostermeier, A. Completely derandomized self-adaptation in evolution strategies. *Evol. Comput.*, 9 (2):159–195, 2001.

Hennig, P. and Schuler, C.J. Entropy search for information-efficient global optimization. *The Journal of Machine Learning Research*, 98888:1809–1837, 2012.

Hoffman, M., Kueck, H., de Freitas, N., and Doucet, A. New inference strategies for solving Markov decision processes using reversible jump MCMC. In *Uncertainty in Artificial Intelligence*, pp. 223–231, 2009.

Hoffman, M., Brochu, E., and de Freitas, N. Portfolio allocation for Bayesian optimization. In *Uncertainty in Artificial Intelligence*, pp. 327–336, 2011.

Hoffman, M.W., Shahriari, B., and de Freitas, N. On correlation and budget constraints in model-based multi-armed-bandit optimization with application to automatic machine learning. In *AI and Statistics*, 2014.

Hutter, F., Hoos, H. H., and Leyton-Brown, K. Sequential model-based optimization for general algorithm configuration. In *Proceedings of LION-5*, pp. 507523, 2011.

Jones, D. R., Perttunen, C. D., and Stuckman, B. E. Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1): 157–181, 1993.

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

Kaufmann, E., Korda, N., and Munos, R. Thompson sampling: An asymptotically optimal finite-time analysis. In *Algorithmic Learning Theory*, volume 7568 of *Lecture Notes in Computer Science*, pp. 199–213. Springer Berlin Heidelberg, 2012.

Kim, J., Ohta, T., Tateisi, Y., and ichi Tsujii, J. GENIA corpus - a semantically annotated corpus for bio-textmining. In *ISMB (Supplement of Bioinformatics)*, pp. 180–182, 2003.

Kocsis, L. and Szepesvári, C. Bandit based Monte-Carlo planning. In *European Conference on Machine Learning*, pp. 282–293. 2006.

Kueck, H., de Freitas, N., and Doucet, A. SMC samplers for Bayesian optimal nonlinear design. In *IEEE Nonlinear Statistical Signal Processing Workshop*, pp. 99–102, 2006.

Kueck, H., Hoffman, M., Doucet, A., and de Freitas, N. Inference and learning for active sensing, experimental design and control. In Araujo, H., Mendonca, A., Pinho, A., and Torres, M. (eds.), *Pattern Recognition and Image Analysis*, volume 5524, pp. 1–10. Springer Berlin Heidelberg, 2009.

Lizotte, D., Greiner, R., and Schuurmans, D. An experimental methodology for response surface optimization methods. *J. of Global Optimization*, pp. 1–38, 2011.

Mahendran, N., Wang, Z., Hamze, F., and de Freitas, N. Adaptive MCMC with Bayesian optimization. *Journal of Machine Learning Research - Proceedings Track*, 22: 751–760, 2012.

Martinez-Cantin, R., de Freitas, N., Doucet, A., and Castellanos, J. A. Active policy learning for robot planning and exploration under uncertainty. *Robotics Science and Systems*, 2007.

May, B. C., Korda, N., Lee, A., and Leslie, D. S. Optimistic Bayesian sampling in contextual bandit problems. Technical Report 11:01, Statistics Group, School of Mathematics, University of Bristol, 2011.

Močkus, J. The Bayesian approach to global optimization. In *Systems Modeling and Optimization*, volume 38, pp. 473–481. Springer, 1982.

Munos, R. Optimistic optimization of a deterministic function without the knowledge of its smoothness. In *Advances in Neural Information Processing Systems*, pp. 783–791, 2011.

Munos, R. From Bandits to Monte-Carlo Tree Search: The Optimistic Principle Applied to Optimization and Planning. Technical Report hal-00747575, INRIA Lille, 2014.

Parameswaran, A., Garcia-Molina, H., and Rajaraman, A. Towards the web of concepts: Extracting concepts from large datasets. *Proceedings of the VLDB Endowment*, 3 (1-2):566–577, 2010.

Rasmussen, C. E. and Williams, C. K. I. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

Snoek, J., Larochelle, H., and Adams, R. P. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, 2012.

Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M. Gaussian process optimization in the bandit setting: No regret and experimental design. In *International Conference on Machine Learning*, pp. 1015–1022, 2010.

Stein, M. L. *Interpolation of Spatial Data: Some Theory for Kriging*. Springer, 1999.

Valko, M., Carpentier, A., and Munos, R. Stochastic simultaneous optimistic optimization. In *International Conference on Machine Learning*, 2013.

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

Wang, Z., Zoghi, M., Matheson, D., Hutter, F., and de Freitas, N. Bayesian optimization in high dimensions via random embeddings. In *International Joint Conference on Artificial Intelligence*, 2013.