Local Bayesian Optimization of Motor Skills

Riad Akrour [1] **Dmitry Sorokin** [1] **Jan Peters** [1 2] **Gerhard Neumann** [1 3]

Abstract

Bayesian optimization is renowned for its sample efficiency but its application to higher dimensional tasks is impeded by its focus on global optimization. To scale to higher dimensional problems, we leverage the sample efficiency of Bayesian optimization in a local context. The optimization of the acquisition function is restricted to the vicinity of a Gaussian search distribution which is moved towards high value areas of the objective. The proposed information-theoretic update of the search distribution results in a Bayesian interpretation of local stochastic search: the search distribution encodes prior knowledge on the optimum's location and is weighted at each iteration by the likelihood of this location's optimality. We demonstrate the effectiveness of our algorithm on several benchmark objective functions as well as a continuous robotic task in which an informative prior is obtained by imitation learning.

1. Introduction

Recent advances in deep reinforcement learning, supported by the ability of generating and processing large amounts of data, allowed impressive achievements such as playing Atari at human level (Mnih et al., 2015) or mastering the game of Go (Silver et al., 2016). In robotics however, sample complexity is paramount as sample generation on physical systems cannot be sped up and can cause wear and damage to the robot when excessive (Kober et al., 2013). Relying on a simulator to carry the learning will inevitably result in a reality gap, since mechanical forces such as stiction are hard to accurately model. However, a policy learned in a simulated environment can still be valuable provided the availability of a sample efficient algorithm to carry an additional optimization phase on the physical system.

Bayesian optimization is best known as a black-box global optimizer (Brochu et al., 2010; Shahriari et al., 2016). It was shown to be efficient for several function landscapes (Jones, 2001), real world scenarios such as the automatic tuning of machine learning algorithms (Bergstra et al., 2011; Snoek et al., 2012; Feurer et al., 2015) or robotics and control (Lizotte et al., 2007; Wilson et al., 2014; Calandra et al., 2016) and several of its variants have convergence guaranties to a global optimum (Vazquez & Bect, 2010; Bull, 2011). Its efficiency stems from two key principles: a probabilistic modeling of the objective function and a sampling procedure that fully exploits this model. However, as the dimensionality of the task increases, non-stationarity effects of the objective or the noise function (see Shahriari et al. (2016), Sec. V.D. for a discussion of these effects) are exacerbated, rendering the modeling of the objective function challenging. An additional difficulty stemming from the increase in dimensionality is the tendency of Bayesian optimization to over-explore, which was experimentally observed in e.g. Brochu et al. (2007). Several recent approaches trying to scale Bayesian optimization to higher dimensions assume additional structure of the objective function. In Snoek et al. (2014), it is assumed that stationarity of the objective function can be recovered through the use of a parametric family of mappings. While it is assumed that the objective function has a lower intrinsic dimension in Djolonga et al. (2013); Wang et al. (2016), can be decomposable into a sum of lower dimensional functions in Kandasamy et al. (2015) or a combination of both hypothesis in Li et al. (2016).

In this paper, we assume prior knowledge on the location of the optimum—given by an initial solution and a confidence on the optimality thereof—and leverage Bayesian optimization in a local manner to improve over this solution. We are especially interested in the application of our algorithm to the optimization of motor skills since i) evaluating the policy return is expensive on physical systems and will likely dominate the computational budget of the optimization process; as such, sample efficient algorithms such as Bayesian optimization are desirable ii) robotics applications are typically high dimensional and global optimization might be prohibitively expensive iii) an initial solution

[1]CLAS/IAS, TU Darmstadt, Darmstadt, Germany [2]Max Planck Institute for Intelligent Systems, Tübingen, Germany [3]L-CAS, University of Lincoln, Lincoln, United Kingdom. Correspondence to: Riad Akrour <riad@robot-learning.de>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017.

can often be obtained through the use of imitation learning (Argall et al., 2009) or by a preliminary optimization on a surrogate model such as a simulator.

2. Related work

Our algorithm can be seen as a local stochastic search algorithm akin to Covariance Matrix Adaptation (CMA-ES) (Hansen & Ostermeier, 2001), cross-entropy (Mannor et al., 2003) or MOdel-based Relative Entropy (MORE) (Abdolmaleki et al., 2015). Local stochastic search algorithms typically maintain a Gaussian search distribution from which samples are generated, the objective function is evaluated and the search distribution is updated. As in Bayesian optimization, they are of particular use when the gradient of the objective function is unknown. Their use as a black-box optimization routine is gaining popularity in the machine learning community, e.g. in reinforcement learning (Thomas et al., 2015) or even for hyperparameter tuning (Bergstra et al., 2011) and the optimization of the acquisition function (Wang et al., 2016) of global Bayesian optimization.

Our algorithm shares the same general structure as local stochastic search algorithms and additionally learns a (probabilistic) model of the objective function. Modeling the objective function was already explored in the stochastic search literature. A surrogate function is learned in (Loshchilov et al., 2013) using SVM-Rank, and is optimized using CMA-ES for a few iterations, yielding an update of the search distribution without requiring additional function evaluations. While in MORE (Abdolmaleki et al., 2015), a local quadratic approximation of the objective function yields the new mean and covariance of the Gaussian search distribution upon an information-theoretic update. Unlike these algorithms, we do not optimize the learned (probabilistic) model, but derive from it $p(\boldsymbol{x} = \boldsymbol{x}^\star|\mathcal{D})$, the probability of $\boldsymbol{x}$ being optimal. Our search distribution is then updated such as to minimize the Kullback-Leibler (KL) divergence to $p(\boldsymbol{x} = \boldsymbol{x}^\star|\mathcal{D})$. Compared to these surrogate assisted local stochastic search algorithms (Loshchilov et al., 2013; Abdolmaleki et al., 2015), the transformation of the optimization landscape (minimizing the KL-divergence to $p(\boldsymbol{x} = \boldsymbol{x}^\star|\mathcal{D})$ instead of the objective function) facilitates learning of the surrogate model by lowering the variance in poorly performing regions, as illustrated in Fig. 1.

To approximate $p(\boldsymbol{x} = \boldsymbol{x}^\star|\mathcal{D})$ we rely on a probabilistic modeling of the objective function and to select the next point to sample we locally optimize an acquisition function. As such, our algorithm can also be seen as Bayesian optimization where the usual box constraint is moved towards a high value area of the objective function to restrict exploration.

Algorithm 1 Local Bayesian Optimization of Motor Skills

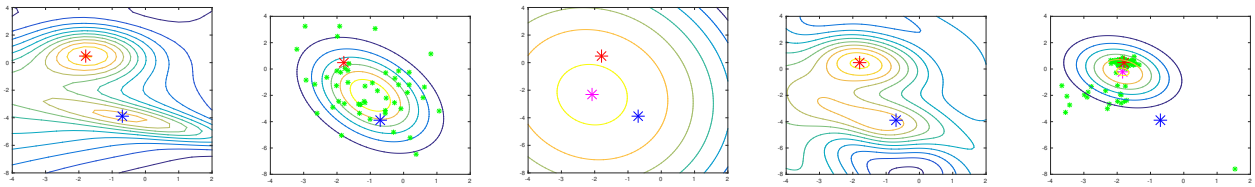
Input: Initial policy $\pi_0 = \mathcal{N}(\boldsymbol{\mu}_0, \sigma_0^2 I)$, step-size ϵ, entropy reduction rate β
Output: Policy π_N
for $n = 1$ **to** N **do**
  Fit: Gaussian $\widehat{p}_n$ from local samples of $p_n^\star$ (Sec. 3.2)
  Optimize: $(\eta^*, \omega^*) = \arg\min g_n(\eta, \omega)$ (Sec. 3.1.1)
  Bayesian Update: $(\pi_{n+1})^{\eta^\star+\omega^\star} \propto \pi_n^{\eta^\star} p_n^\star$ (Sec. 3.1.1)
  Evaluate: $\boldsymbol{x}_n$ from local samples of $p_n^\star$ (Sec. 3.3)
  $\mathcal{D}_n \longleftarrow \mathcal{D}_{n-1} \cup \{(\boldsymbol{x}_n, y_n)\}$
end for

In reinforcement learning, probabilistic modeling was used to e.g. learn a transition model (Deisenroth & Rasmussen, 2011) or the policy gradient (Ghavamzadeh et al., 2016) with Gaussian processes. Closer to our work, the use of an adaptive box constraint was explored in Bayesian optimization to ensure a safe optimization of a robot controller (Berkenkamp et al., 2016; Englert & Toussaint, 2016). Considering safety is crucial for motor skill learning on physical systems to prevent the evaluation of 'dangerous' parameters. Both approaches restrict exploration to an initial safe region of the parameter space that is incrementally expanded using additional problem assumptions. Without such assumptions our algorithm cannot guarantee safety but its local nature is expected to dampen the potential risk of global Bayesian optimization.

3. Local Bayesian optimization

Let $f\colon \mathbb{R}^d \mapsto \mathbb{R}$ be an objective function. For example $f(\boldsymbol{x})$ can be the expected reward of a robot controller parameterized by $\boldsymbol{x} \in \mathbb{R}^d$. We assume that the algorithm only has access to noisy evaluations $y = f(\boldsymbol{x}) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma_s^2)$ is Gaussian noise of unknown deviation σ_s. The algorithm will produce a sequence $\{(\boldsymbol{x}_1, y_1) \dots (\boldsymbol{x}_N, y_N)\}$ of parameter-evaluation pairs and the goal is to minimize the cumulative regret $\frac{1}{N}\sum_i f(\boldsymbol{x}^\star) - y_i$ for some global maximizer $\boldsymbol{x}^\star$ of f. The cumulative regret emphasizes the inherent cost in evaluating a bad parameter, potentially causing wear and damage to the robot.

Prior knowledge on an optimum's location $\boldsymbol{x}^\star$ is given to the algorithm by a Gaussian distribution $\pi_0 = \mathcal{N}(\boldsymbol{\mu}_0, \sigma_0^2 \boldsymbol{I})$. In what follows we will indistinctly refer to π as a search distribution or a policy following the terminology of the stochastic search and reinforcement learning (RL) communities. In an RL context, an informative prior can often be obtained from human generated data or from a simulator. Specifically, we assume that the mean $\boldsymbol{\mu}_0$ of π_0 is obtained by imitation learning if near-optimal demonstrations are available or by a preliminary optimization on a less accurate but inexpensive model of the system dy-



(a) Objective function f. (b) Search distribution. (c) Quad. approx. of f. (d) GP approx. of f. (e) Gaussian approx. of $p(\boldsymbol{x}^\star|\mathcal{D})$.

Figure 1. The benefits of minimizing the KL divergence to $p(\boldsymbol{x}^\star|\mathcal{D})$ instead of maximizing a local approximation of the objective function f. a) The objective function f has two modes illustrated by the red (higher mode) and blue (lower mode) stars. b) The search distribution π draws samples in the area of both modes. The samples and their evaluation are stored in $\mathcal{D}$. c) The quadratic approximation of f (minimizing the empirical mean squared error on $\mathcal{D}$) captures variations of f even in low value areas and result in a poor orientation for the search distribution update as illustrated by the model's optimum (magenta star in (c)). On the other side, the Gaussian process approximation of f (shown in (d)) is confident enough in its approximation for $p(\boldsymbol{x}^\star|\mathcal{D})$ to sample mainly around the higher mode (red star, see (e)). As a result, the Gaussian approximation of $p(\boldsymbol{x}^\star|\mathcal{D})$ only focuses on high value areas and results in a better direction for the search distribution update than the quadratic model of f.

namics. Whereas σ_0 is a hyper-parameter of the algorithm, manually set in our experiments, and expressing the confidence in the optimality of $\boldsymbol{\mu}_0$.

The search distribution π_n is updated by solving the optimization problem formally defined in Sec. 3.1.1. The objective of the optimization problem is to minimize the KL divergence between π_n and $p(\boldsymbol{x} = \boldsymbol{x}^\star|\mathcal{D}_n)$, the probability of $\boldsymbol{x}^\star$ being optimal according to the data set of parameter-evaluation pairs $\mathcal{D}_n$. Solving this problem results in a Bayesian update, as shown in Alg. 1, where the prior $\pi_n(\boldsymbol{x})$ on the optimality of $\boldsymbol{x}$ is weighted by the likelihood $p(\boldsymbol{x} = \boldsymbol{x}^\star|\mathcal{D}_n)$ of $\boldsymbol{x}$ being optimal according to $\mathcal{D}_n$. Letting the likelihood $p(\boldsymbol{x} = \boldsymbol{x}^\star|\mathcal{D}_n)$ be denoted by $p^\star_n(\boldsymbol{x})$, the first step of the algorithm is to fit $\widehat{p}_n$, a Gaussian approximation of $p^\star_n$ (Sec. 3.2). Subsequently, a dual function is optimized (Sec. 3.1.1) to make sure that the search distribution moves slowly towards $p^\star$ as new evaluations are collected. Modulating the Bayesian update with the dual parameters η^* and ω^* is important since $\mathcal{D}_n$ is initially empty and $p^\star_n$ not initially informative. Finally, a new evaluations of f is requested by selecting $\boldsymbol{x}_n$ from the previously generated samples of $p^\star_n$ and the process is iterated.

The next subsections give a detailed presentation of both the search distribution update and the sampling procedure from $p^\star_n$.

3.1. Search distribution update

The search distribution in our algorithm is updated such as to minimize the KL divergence between π_n and $p^\star_n$. The resulting optimization problem is closely related to the one solved by the MORE algorithm (Abdolmaleki et al., 2015). In the next subsections, we will first formalize our search distribution update before briefly describing the search distribution update of MORE and showing how their derivations can be used to obtain our search distribution update.

3.1.1. The optimization problem

The search distribution is updated such that its KL divergence w.r.t. $p^\star_n$ is minimized. Since future evaluations of f will be performed around the updated search distribution, it becomes critical to control the change of distribution between iterations by constraining the aforementioned minimization problem. These constraints will ensure that the exploration is not reduced too fast or that the mean is not moved too quickly from the initial solution $\boldsymbol{\mu}_0$. The resulting optimization problem is given by

$$\begin{aligned} \arg\min_{\pi} \quad & \mathrm{KL}(\pi \parallel p^\star_n), \\ \text{subject to} \quad & \mathrm{KL}(\pi \parallel \pi_n) \leq \epsilon, \quad (1) \\ & \mathcal{H}(\pi_n) - \mathcal{H}(\pi) \leq \beta, \quad (2) \end{aligned}$$

where $\mathrm{KL}(p \parallel q) = \int p(\boldsymbol{x}) \log \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})} d\boldsymbol{x}$ is the KL divergence between p and q and $\mathcal{H}(p) = -\int p(\boldsymbol{x}) \log(p(\boldsymbol{x})) d\boldsymbol{x}$ is the entropy of p. The hyper-parameters ϵ and β respectively bound the change in distribution and the reduction in entropy between successive iterations.

The use of the KL divergence to constrain the update is widespread in the reinforcement learning community (Peters et al., 2010; Schulman et al., 2015). When the search distributions π and π_n are of Gaussian form, the KL divergence in Eq. (1) is impacted by three factors. On one side, by the change in entropy between the two distributions—having a direct impact on the exploration rate. On the other side, by the displacement of the mean and the rotation of the covariance matrix—not impacting the exploration rate. To better control the exploration, we choose to decouple

the reduction in entropy from the KL constraint. It was shown in (Abdolmaleki et al., 2015) that the additional entropy constraint can lead to significantly better solutions at the expense of a slower start.

The optimization problem defined in this section is closely related to the one solved by MORE. In fact, when the inequality (2) is replaced by the equality constraint $\mathcal{H}(\pi_n) - \mathcal{H}(\pi) = \beta$ for both algorithms then the two problems coincide; while only a small modification of the dual function is necessary otherwise. For the sake of clarity and to keep the paper self-contained, we will briefly introduce MORE before showing how we can reuse their derivation of the search distribution update in our algorithm.

3.1.2. Model-Based Relative Entropy Search

MORE (Abdolmaleki et al., 2015) is a local stochastic search algorithm where the search distribution $\pi_n(\boldsymbol{x})$ is updated by solving the following constrained problem

$$\begin{aligned} \arg\max_{\pi} \quad & \int \pi(\boldsymbol{x}) f(\boldsymbol{x}) d\boldsymbol{x} \\ \text{subject to} \quad & \mathrm{KL}(\pi \parallel \pi_n) && \leq \epsilon, \quad (3) \\ & \mathcal{H}(\pi_n) - \mathcal{H}(\pi) && \leq \beta, \quad (4) \end{aligned}$$

An analytic solution of the problem is obtained by locally approximating f with the quadratic model

$$R_n(\boldsymbol{x}) = -\frac{1}{2}\boldsymbol{x}^T \mathbf{R}_n \boldsymbol{x} + \boldsymbol{x}^T \boldsymbol{r}_n + r_n,$$

learned by linear regression from the data set $\mathcal{D}_n$. Letting the search distribution $\pi_n(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$ at iteration n be parameterized by the mean $\boldsymbol{\mu}_n$ and covariance matrix $\boldsymbol{\Sigma}_n$, the aforementioned optimization problem yields the closed form update where the new mean and covariance are given by

$$\boldsymbol{\Sigma}_{n+1}^{-1} = (\eta^\star + \omega^\star)^{-1} \left(\eta^\star \boldsymbol{\Sigma}_n^{-1} + \boldsymbol{R}_n\right), \quad (5)$$

$$\boldsymbol{\mu}_{n+1} = (\eta^\star + \omega^\star)^{-1} \boldsymbol{\Sigma}_{n+1} \left(\eta^\star \boldsymbol{\Sigma}_n^{-1} \boldsymbol{\mu}_n + \boldsymbol{r}_n\right), \quad (6)$$

where $\eta^\star$ and $\omega^\star$ are the Lagrange multipliers of the constraints (3) and (4) respectively, and are obtained by minimizing the dual function $g_n(\eta, \omega)$ by gradient descent (Abdolmaleki et al., 2015).

As can be seen in Eq. 5, the new covariance matrix is a trade-off between the old covariance and the local curvature of the objective function f—where the trade-off parameters are computed in order to satisfy both constraints of the optimization problem. As such, it is appropriate to use the covariance matrix of the search distribution in the kernel function for the local approximation of f when using GP regression.

We additionally define MORE with equality constraint as a variant of MORE where the inequality constraint in (4) is replaced with the equality constraint $\mathcal{H}(\pi_n) - \mathcal{H}(\pi) = \beta$, forcing the reduction in entropy at each iteration to be exactly β. This modification will not change the shape of the update but only the Lagrange multipliers, that can be obtained by simply alleviating the constraint $\omega \geq 0$ in the minimization of $g_n(\eta, \omega)$.

3.1.3. Optimization problems' equivalence

We now show that the optimization problem in our algorithm can be phrased as the optimization problem solved by MORE for the equality entropy constraint; while only a small modification of the dual minimization is required for the inequality entropy constraint. The equivalence of the optimization problems will allow us to use Eq. 5 and 6 to update our search distribution.

Proposition 1. *The optimization problem in Sec. 3.1.1 can be reduced to the optimization problem in 3.1.2 for the objective function $f = \log p_n^\star$ when both problems enforce an exact entropy reduction constraint on π.*

Proof. We first rephrase the problem in Sec. 3.1.1 as the maximization over π of

$$-\mathrm{KL}(\pi \parallel p_n^\star) + \beta - \mathcal{H}(\pi_n),$$

where we switched the sign of the KL divergence term and added the constant term $\beta - \mathcal{H}(\pi_n)$. These modifications will not change the value of the stationary points of the Lagrangian. The resulting Lagrangian is

$$\begin{aligned} \mathcal{L}(\pi, \eta, \omega) = \int \pi(\boldsymbol{x}) \log p_n^\star(\boldsymbol{x}) d\boldsymbol{x} + \eta(\epsilon - \mathrm{KL}(\pi \parallel \pi_n)) \\ + (\omega + 1)(\mathcal{H}(\pi) - \mathcal{H}(\pi_n) + \beta), \end{aligned}$$

with dual variables $\eta \geq 0$ and $\omega \in \mathbb{R}$ and where we have split the term $\mathrm{KL}(\pi \parallel p_n^\star)$ into the expected log-density of $p_n^\star$ and the entropy $\mathcal{H}(\pi)$ of π. A MORE formulation with similar entropy and KL divergence constraints and where the objective is to maximize the log-density $\log p_n^\star$ yields the Lagrangian

$$\begin{aligned} \mathcal{L}'(\pi, \eta, \omega) = \int \pi(\boldsymbol{x}) \log p_n^\star(\boldsymbol{x}) d\boldsymbol{x} + \eta(\epsilon - \mathrm{KL}(\pi \parallel \pi_n)) \\ + \omega(\mathcal{H}(\pi) - \mathcal{H}(\pi_n) + \beta). \end{aligned}$$

Since we have no constraint on ω, it is easy to see that the dual variable minimizing the dual of the first problem $\omega^\star$ and of the second (MORE) problem ω'^* are related by $\omega^\star = \omega'^* - 1$ and both problems will result in the same update of π. □

Intuitively, the minimization of $\mathrm{KL}(\pi \parallel p_n^\star)$ can be reduced to the maximization (in expectation of π) of the log-density

$\log p_n^\star$ because the equality constraint $\mathcal{H}(\pi) = \mathcal{H}(\pi_n) - \beta$ annihilates the effect of the additional entropy term $\mathcal{H}(\pi)$ coming from the KL objective.

From Proposition 1 and following the derivations in (Abdolmaleki et al., 2015), the search distribution π_{n+1} solution of the optimization problem in Sec. 3.1.1 is given by

$$\pi_{n+1} \propto (\pi_n)^{\frac{\eta^\star}{\eta^\star+\omega^\star}} \, (p_n^\star)^{\frac{1}{\eta^\star+\omega^\star}} \, , \tag{7}$$

where $\eta^\star$ and $\omega^\star$ are the Lagrange multipliers related to the KL and entropy constraints respectively and minimizing the dual function $g_n(\eta, \omega)$. We refer the reader to Sec. 2.1 in (Abdolmaleki et al., 2015) for the definition of $g_n(\eta, \omega)$.

When the entropy constraint is the inequality in (2) instead of an equality, the Lagrange multipliers for our update and the MORE update may differ. However, $\eta^\star$ and $\omega^\star$ can still be obtained by the minimization of the same $g_n(\eta, \omega)$ with the additional constraint $\omega \geq 1$.

Note that the new search distribution π_{n+1} as defined in Eq. (7) is not necessarily Gaussian because of the multiplication by $p_n^\star$. However, by approximating $p_n^\star$ by a Gaussian distribution $\widehat{p}_n$, $\log \widehat{p}_n$ will be a quadratic model and Eq. 5 and 6 can be used to obtain a Gaussian π_{n+1}.

3.2. Approximating the argmax distribution

To obtain a closed form update of the Gaussian search distribution in Eq. (7), we will approximate $p_n^\star$ by fitting a Gaussian $\widehat{p}_n$ to samples of $p_n^\star$ as shown in Fig. 1e. To generate samples from $p_n^\star$, we use Thompson sampling (Chapelle & Li, 2011; Russo & Roy, 2014) from a probabilistic model of the objective function f.

The probabilistic model of f follows from both a Gaussian process (GP) prior and a Gaussian likelihood assumption. We use in this paper the squared exponential kernel $k_n(\boldsymbol{x}_i, \boldsymbol{x}_j) = \theta_0 \exp(-\theta_1(\boldsymbol{x}_i - \boldsymbol{x}_j)^T \Sigma_n^{-1}(\boldsymbol{x}_i - \boldsymbol{x}_j))$ with hyper-parameters θ_0 and θ_1 and Σ_n the covariance matrix of π_n. The resulting model has hyper-parameter vector $\boldsymbol{\phi} = (\theta_0, \theta_1, \sigma_s)$, where σ_s^2 is the noise variance of the likelihood function as previously defined. Samples from $p_n^\star$ are generated by i) sampling a hyper-parameter vector $\boldsymbol{\phi}$ from the posterior distribution $p(\boldsymbol{\phi}|\mathcal{D}_n)$ using slice sampling (Murray & Adams, 2010), ii) sampling a function from the GP posterior $p(\widetilde{f}|\mathcal{D}_n, \boldsymbol{\phi})$ and iii) returning the argmax of $\widetilde{f}$.

The computational complexity of evaluating $\widetilde{f}$ is cubical in the number of requested evaluations as it involves a matrix inversion. As such, the exact maximization of $\widetilde{f}$ can prove to be challenging. Prior work in the literature considered approximating $\widetilde{f}$ with a linear function (see for example Hernández-Lobato et al. (2014), Sec. 2.1) and globally maximizing the linear surrogate. In our local optimization context, we follow a more straightforward approach by generating samples from π_n, and returning the sample with maximal value of $\widetilde{f}$. The rational behind searching the argmax of $\widetilde{f}$ in the vicinity of π_n is that samples from π_n are likely to have high $\widetilde{f}$ value since π_n is updated such that the KL divergence w.r.t. $p_n^\star$ is minimized. The repeated process of drawing points from π_n, drawing their value from the GP posterior and selecting the point with highest value will constitute a data set $\mathcal{D}_n^\star$ containing local samples from $p_n^\star$.

Once samples from $p_n^\star$ are generated and stored in $\mathcal{D}_n^\star$, we set $\widehat{p}_n = \mathcal{N}(\boldsymbol{\mu}_n^\star, \Sigma_n^\star)$ where $\boldsymbol{\mu}_n^\star$ and $\Sigma_n^\star$ are the sample mean and covariance of the samples in $\mathcal{D}_n^\star$. Because $\widehat{p}_n$ is Gaussian, $\log \widehat{p}_n$ is quadratic and the search distribution update in Eq. (7) yields a Gaussian distribution π_{n+1} with covariance and mean as defined in Eq. (5) and Eq. (6) respectively with $\boldsymbol{R}_n = {\Sigma_n^\star}^{-1}$ and $\boldsymbol{r}_n = {\Sigma_n^\star}^{-1}\boldsymbol{\mu}_n^\star$.

3.3. Sample generation

The function f is initially evaluated at a point $\boldsymbol{x}_0$ drawn from the prior distribution π_0. In subsequent iterations, a point $\boldsymbol{x}_n$ is randomly selected from $\mathcal{D}_n^\star$, the set of samples used in the computation of $\widehat{p}_n$ (Sec. 3.2).

Experimentally, we noticed that the exploration in our algorithm is heavily influenced by the centering of the values $\{y_i\}_1^n$ in $\mathcal{D}_n$. Three variants of our algorithm are initially evaluated with different target values of the GP. The target values are obtained by subtracting from y_i either the $\max$ the $\min$ or the mean of $\{y_i\}_1^n$. Since the GP modeling of f has a zero mean prior, the extreme case where the $\max$ (resp. the $\min$) is subtracted from the data results in an optimistic (resp. pessimistic) exploration strategy considering that the objective function in unexplored areas have values higher (resp. lower) in expectation than the best (resp. worst) evaluation so far.

4. Experiments

We initially investigate in this section the impact of the target centering (Sec. 3.3) on the exploration-exploitation trade-off of our algorithm. We then compare our algorithm to two state-of-the-art model based optimizers: the global Bayesian optimizer and the local Model-Based Relative Entropy Search (Abdolmaleki et al., 2015). The algorithms are compared on several continuous function benchmarks as well as a simulated robotics task.

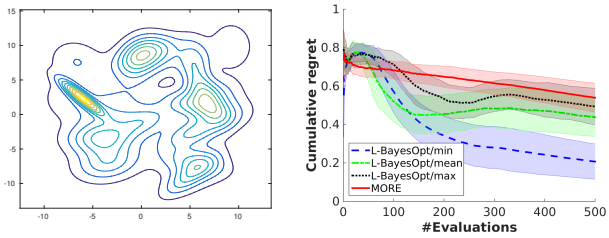
Benchmarks. Variants of our algorithm are first compared on randomly generated smooth 2 dimensional objective functions. We then conduct a comparison to the state-of-the-art on the COmparing COntinuous optimisers (COCO)

testbed on the 20 functions $f5$ to $f24$ (we refer the reader to http://coco.gforge.inria.fr/ for an illustration and the mathematical definition of each function). We chose to split the experimentation between the uni-modal and the multi-modal categories of the testbed. The uni-modal category is representative of the informed initialization hypothesis that only requires local improvements. While the multi-modal category assesses the robustness of our algorithm to more complex function landscapes—which can be encountered in practice despite the informed initialization if e.g. a too wide variance σ_0^2 is initially set. We vary the dimension of the COCO functions from 3 to 30 while the robotics task evaluates our algorithm on a 70 dimensional setting.

Algorithms. In what follows, we will refer to our algorithm as L-BayesOpt. We rely on the GPStuff library (Vanhatalo et al., 2013) for the GP implementation and the posterior sampling of hyper-parameters. We use the BayesOpt library (Martinez-Cantin, 2014) for global Bayesian optimization with a similar to L-BayesOpt squared exponential kernel and MCMC sampling of hyper-parameters and an additional Automatic Relevance Determination step executed every 50 samples. In the experiments we evaluate BayesOpt with both Expected Improvement and Thompson Sampling acquisition functions.

In all of the experiments, L-BayesOpt and MORE will share the same initial policy, step-size ϵ, entropy reduction β and will sample ten points per iteration. We choose to use an equality constraint for the entropy reduction for both algorithms. As a result, both L-BayesOpt and MORE will have the same entropy at every iteration and any difference in performance will be attributed to a better location of the mean, adaptation of the covariance matrix or sampling procedure rather than a faster reduction in exploration. In all but the last experiment $\epsilon = \beta = .05$ while for the robotics experiment with an initial solution learned by imitation learning we set a more aggressive step size and entropy reduction $\epsilon = \beta = 1$.

Evaluation criterion. The performance metric in RL is typically given by the average return $J(\pi_n) = \int \pi_n(\boldsymbol{x}) f(\boldsymbol{x}) d\boldsymbol{x}$ while in Bayesian optimization it is typically determined by the minimal evaluation $\min_{1 \leq i \leq n} y_i$ reached at iteration n. When the evaluations are noisy the minimum evaluation is not a robust performance metric—nor an appropriate criterion for the algorithm to select the returned optimizer. In order to have a common evaluation criterion, all the approaches are seen as multi-armed bandit algorithms and we use the cumulative regret $\frac{1}{n} \sum_i f(\boldsymbol{x}^\star) - y_i$ as the evaluation criterion. The cumulative regret of (global) Bayesian optimizers is expected to be asymptotically lower than that of local optimizers as it always finds the global maximum given sufficiently many evaluations.



(a) Sample objective function. (b) Cumulative regret.

Figure 2. Three variants of L-BayesOpt (Sec. 3.3) and MORE evaluated on 11 randomly generated objective functions. The min variant results in a contained exploration and has the lowest cumulative regret during the first 500 function evaluations.

Conversely, trading-off global optimality for fast local improvements might result in a lower regret for local optimizers when the evaluation budget is moderate.

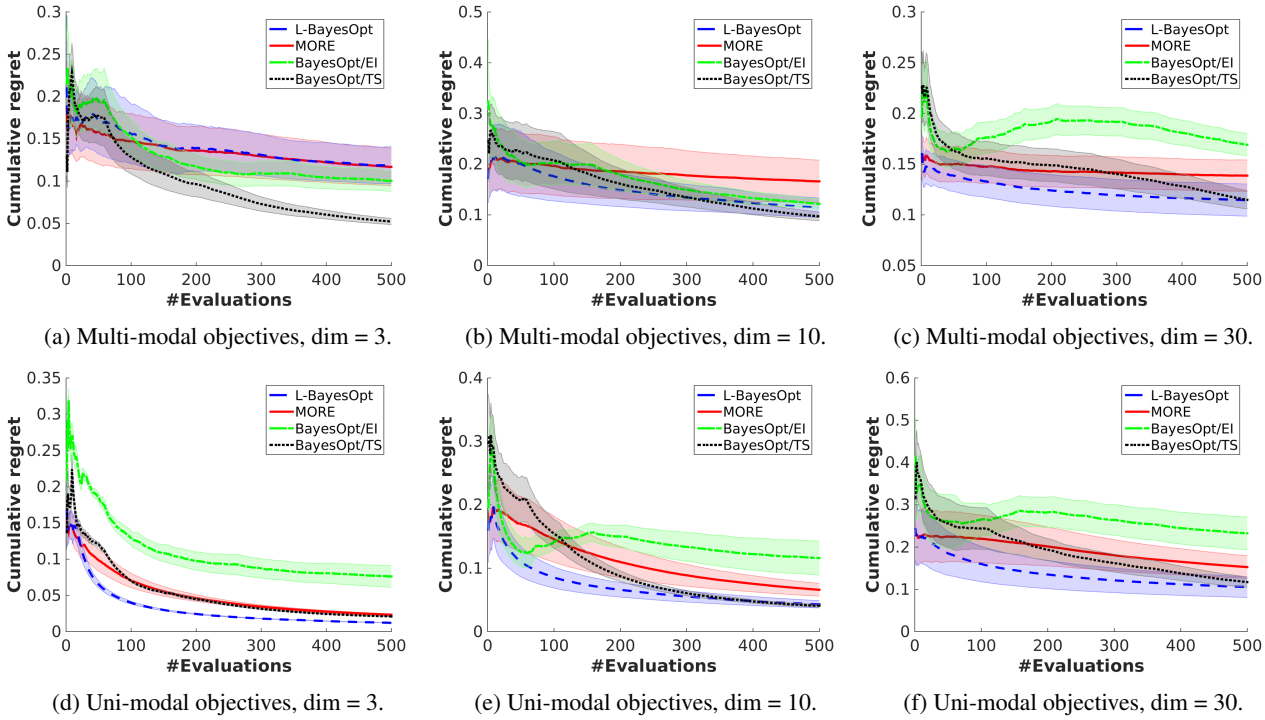
4.1. Exploration variants

In this first set of experiments, we evaluate the different exploration strategies resulting from three different centering methods of the y values in $\mathcal{D}_n$. We compare these three variants of L-BayesOpt on 11 randomly generated two dimensional Gaussian mixture objective functions (see Fig. 2a for an illustration). We chose these functions as they are cheap to evaluate, easy to approximate by a GP and their multi-modal nature is appropriate for evaluating the exploration-exploitation trade-off of the three variants.

As hypothesized in Sec. 3.3, the cumulative regret in Fig. 2b shows that the min variant exhibits the lowest exploration and reduces the regret faster than the other optimizers. Yet, when compared to MORE it manages to converge to better local optima in 5 out of the 11 randomly generated objectives while MORE converges to a better optimum in one of the 6 remaining objectives. Note that MORE manages to decrease the regret faster than our algorithm during the first 100 evaluations. However, the sampling scheme relying on the Thompson sampling acquisition function and the convergence to higher modes gives the advantage to the L-BayesOpt variants after the initial 100 evaluations. In the remainder of the experimental section only the min variant of our algorithm will be considered.

4.2. State-of-the-art benchmark comparisons

We compare our algorithm to MORE and Bayesian optimization on the COCO testbed. We form two sets each containing 10 objective functions. The first one includes uni-modal functions ($f5$ to $f14$) while the second one includes multi-modal function with an adequate ($f15$ to $f19$) and a weak ($f20$ to $f24$) global structure. Each function has a global optimum in $[-5, 5]^D$, where D is the dimension of the objective function that we vary in the set $\{3, 10, 30\}$.



(a) Multi-modal objectives, dim = 3. (b) Multi-modal objectives, dim = 10. (c) Multi-modal objectives, dim = 30.

(d) Uni-modal objectives, dim = 3. (e) Uni-modal objectives, dim = 10. (f) Uni-modal objectives, dim = 30.

Figure 3. Evaluation on uni-modal and multi-modal functions of varying dimension of the COCO testbed of four algorithms: L-BayesOpt, MORE and global Bayesian optimization using either Expected Improvement (EI) or Thompson Sampling (TS) acquisition function. Bayesian optimization is a global optimizer and its combination with Thompson sampling is as such a zero regret algorithm. However, when evaluation budget is moderate ($\leq$ 500), the local optimization performed by L-BayesOpt yields faster improvements than Bayesian optimization when the objective function is uni-modal or when the dimensionality of the problem increases.

The bounding box $[-5, 5]^D$ is provided to Bayesian optimization while for the local stochastic search algorithms we set the initial distribution to $\pi_0 = \mathcal{N}(0, 3I)$. Note that this is not an informed initialization and none of the functions had their optimum on the null vector.

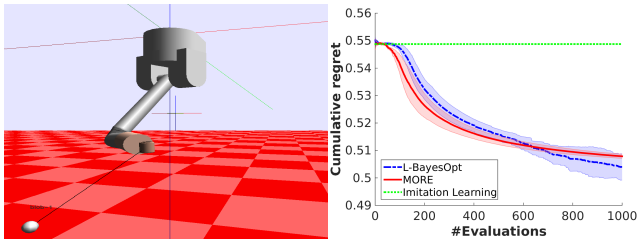
Fig. 3 shows the performance of the four algorithms on the multi-modal (top row) and uni-modal (bottom row) function sets. On the multi-modal set of functions and when $D = 3$, Bayesian optimization with Thompson sampling proves to be an extremely efficient bandit algorithm for uncovering the highest reward point with a minimal number of evaluations. On the contrary, both local stochastic search algorithms struggle to improve over the initial performance. Upon closer inspection, this appears to be especially true for functions with weak global structure such as $f23$. We hypothesize that for these highly multi-modal functions, both model based stochastic search algorithms learn poor quadratic models (when either approximating f or $p_n^\star$). The performance gap between Bayesian optimization and our algorithm reduces however as the dimensionality of the problem increases.

On the uni-modal functions set, our algorithm reduces significantly faster the regret than Bayesian optimization. As the dimension of the objective function increases from $D = 10$ to $D = 30$, more evaluations are required by Bayesian optimization to reach our algorithm. Compared to MORE, and since the objectives are uni-modal, the use of an acquisition function is the main driving factor for the faster decrease of the regret. Note that even if the functions are uni-modal, both local search algorithms are not necessarily zero if the decrease in entropy is too fast.

Both L-BayesOpt and BayesOpt/TS rely on the Thompson sampling acquisition function for selecting the next point to evaluate. While the acquisition function is maximized on the full support of the objective in the case of Bayesian optimization, it is only optimized in the vicinity of the current search distribution by our algorithm. The experiments on the COCO testbed show that when the function landscape enables the learning of an appropriate update direction for moving the search distribution, the adaptive strategy employed by our algorithm can be more efficient than the global search performed by Bayesian optimization.

4.3. Robot ball in the cup

The task's objective is for the Barrett robot arm to swing the ball upward and place it in the cup (Fig. 4a). The



(a) Robot ball in a cup. (b) Cumulative regret.

Figure 4. L-BayesOpt and MORE locally optimizing an imitation learning policy. The 7 DoF robot is controlled by a 70 parameters policy. L-BayesOpt is initially slower but is better at fine-tuning the policy later on. Plots are averaged over 5 runs.

optimization is performed on the 70 weights of the forcing function of a Dynamical Movement Primitive (DMP, Ijspeert & Schaal 2003) controlling the 7 joints of the robot. The initial forcing function weights $\boldsymbol{\mu}_0$ are learned by linear regression from a single demonstrated trajectory that successfully swings the ball up but where the ball lands at circa 20cm from the cup. We compare the performance of MORE and L-BayesOpt in optimizing the initial policy $\pi_0 = \mathcal{N}(\boldsymbol{\mu}_0, I)$ using the same hyper-parameters. The challenge of the task, in addition to the dimension of the action space, stems from the two exploration regimes required by the exploration scheme. While initially a significant amount of noise needs to be introduced to the parameters to get the ball closer to the cup; successfully getting the ball in the cup requires a more careful tuning of the forcing function.

Fig. 4b shows the performance of both MORE and L-BayesOpt on the robot ball in a cup task. MORE has a better initial sample efficiency and gets the ball closer to the cup at a faster pace than L-BayesOpt. However, the acquisition function based sampling scheme of our algorithm was more efficient for discovering parameters that successfully put the ball in the cup and results in a lower regret (averaged over 5 runs) after 1000 evaluations. The experiment shows that for such high dimensional tasks, our algorithm was better at tuning the policy only when the entropy of the search distribution was significantly reduced. This might be due to the low correlation between Euclidean distance between parameters and difference in reward. One promising direction for future work in a reinforcement learning context is to use kernels based on trajectory data distance instead of parameter distance in euclidian space (Wilson et al., 2014).

5. Discussion

The algorithm presented in this paper can be seen as Bayesian optimization where the usual box constraint is rotated, shrunk and moved at each iteration towards the most promising region of the objective function. The constant reduction of the entropy of the search distribution ensures that the objective function is not modeled and optimized on the entirety of its domain. Compared to (global) Bayesian optimization, we experimentally demonstrated on several continuous optimization benchmarks that it results in faster improvements over the initial solution, at the expense of global optimality. This property is especially useful when an initial informative solution is available and only requires to be locally improved.

The computational cost of the search distribution update in our algorithm is significantly higher than most local stochastic search algorithms. This cost mainly arises from the full Bayesian treatment of the modeling of the objective function f. If the evaluation of f is cheap, a better performance per second is obtained by less expensive stochastic search algorithms where the additional computational budget can be spent in running additional randomized restarts of the algorithms (Auger & Hansen, 2005). However, if the optimization cost is dominated by the evaluation of f, the probabilistic modeling proved to be more sample efficient on several benchmarks by actively selecting the next point to evaluate. As a result, when f is expensive to evaluate our algorithm is expected to have better per second performance than state-of-the-art stochastic search algorithms.

The search distribution update proposed in this paper is well founded and results in an interpretable update. At each iteration the current search distribution is simply weighted by $p(\boldsymbol{x} = \boldsymbol{x}^\star|\mathcal{D}_n)$, the probability of $\boldsymbol{x}$ being optimal according to the current data set. Future work can further improve the sample efficiency of our algorithm in at least three ways. First, if the objective function is upper bounded and the bound is known, we expect that the integration of an additional constraint $f(\boldsymbol{x}) < f(\boldsymbol{x}^*)$ for all $\boldsymbol{x}$ to lead to a more accurate probabilistic modeling and a better exploration-exploitation trade-off. Secondly, the search distribution update is phrased as the minimization of the I-projection of $p(\boldsymbol{x} = \boldsymbol{x}^\star|\mathcal{D}_n)$, which has the property of focusing on one mode of the distribution (Bishop, 2006). However, the Gaussian approximation of $p(\boldsymbol{x} = \boldsymbol{x}^\star|\mathcal{D}_n)$ can average over multiple modes if the GP is unsure about which of them is the highest. We expected that a better update direction can be obtained if a clustering algorithm can detect the highest mode from samples of $p(\boldsymbol{x} = \boldsymbol{x}^\star|\mathcal{D}_n)$. Finally and perharps most interestingly, we expect our algorithm to be able to scale to significantly higher dimensional policies in an RL setting if a trajectory data kernel is used (Wilson et al., 2014). Specifically, distance between policies can be measured by the similarity of actions taken in similar states. The local nature of our algorithm will additionally ensure that such similarity is evaluated on states that are likely to be reached by the evaluated policies.

Acknowledgments

The research leading to these results was funded by the DFG Project *LearnRobotS* under the SPP 1527 Autonomous Learning.

References

Abdolmaleki, Abbas, Lioutikov, Rudolf, Peters, Jan R, Lau, Nuno, Pualo Reis, Luis, and Neumann, Gerhard. Model-based relative entropy stochastic search. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 3523–3531. Curran Associates, Inc., 2015.

Argall, Brenna, Chernova, Sonia, Veloso, Manuela M., and Browning, Brett. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57 (5):469–483, 2009.

Auger, Anne and Hansen, Nikolaus. A restart cma evolution strategy with increasing population size. In *IEEE Congress on Evolutionary Computation*, volume 2, pp. 1769–1776. IEEE, 2005.

Bergstra, James, Bardenet, Rémi, Bengio, Yoshua, and Kégl, Balázs. Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 2546–2554, 2011.

Berkenkamp, Felix, Schoellig, Angela P, and Krause, Andreas. Safe controller optimization for quadrotors with gaussian processes. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pp. 491–496. IEEE, 2016.

Bishop, Christopher M. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, 2006.

Brochu, Eric, de Freitas, Nando, and Ghosh, Abhijeet. Active preference learning with discrete choice data. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 409–416, 2007.

Brochu, Eric, Cora, Vlad M., and de Freitas, Nando. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *CoRR*, abs/1012.2599, 2010.

Bull, A. D. Convergence rates of efficient global optimization algorithms. 12:2879–2904, 2011.

Calandra, Roberto, Seyfarth, André, Peters, Jan, and Deisenroth, Marc Peter. Bayesian optimization for learning gaits under uncertainty - an experimental comparison on a dynamic bipedal walker. *Ann. Math. Artif. Intell.*, 76(1-2):5–23, 2016.

Chapelle, Olivier and Li, Lihong. An empirical evaluation of thompson sampling. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 2249–2257. Curran Associates, Inc., 2011.

Deisenroth, M. and Rasmussen, C. PILCO: A Model-Based and Data-Efficient Approach to Policy Search. In *International Conference on Machine Learning (ICML)*, pp. 465–472, 2011.

Djolonga, Josip, Krause, Andreas, and Cevher, Volkan. High-dimensional gaussian process bandits. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 1025–1033, 2013.

Englert, Peter and Toussaint, Marc. Combined optimization and reinforcement learning for manipulations skills. In *Robotics:Science and Systems 2016*, 2016.

Feurer, Matthias, Klein, Aaron, Eggensperger, Katharina, Springenberg, Jost Tobias, Blum, Manuel, and Hutter, Frank. Efficient and robust automated machine learning. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 2962–2970, 2015.

Ghavamzadeh, Mohammad, Engel, Yaakov, and Valko, Michal. Bayesian policy gradient and actor-critic algorithms. *Journal of Machine Learning Research*, 17(66): 1–53, 2016.

Hansen, Nikolaus and Ostermeier, Andreas. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

Hernández-Lobato, José Miguel, Hoffman, Matthew W., and Ghahramani, Zoubin. Predictive entropy search for efficient global optimization of black-box functions. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 918–926, 2014.

Ijspeert, A. and Schaal, S. Learning Attractor Landscapes for Learning Motor Primitives. In *Advances in Neural Information Processing Systems 15*, (NIPS). MIT Press, Cambridge, MA, 2003.

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

Kandasamy, Kirthevasan, Schneider, Jeff G., and Póczos, Barnabás. High dimensional bayesian optimisation and bandits via additive models. In *International Conference on Machine Learning (ICML)*, pp. 295–304, 2015.

Kober, J., Bagnell, J. Andrew, and Peters, J. Reinforcement learning in robotics: A survey. *International Journal of Robotics Research*, July 2013.

Li, Chun-Liang, Kandasamy, Kirthevasan, Póczos, Barnabás, and Schneider, Jeff G. High dimensional bayesian optimization via restricted projection pursuit models. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016, Cadiz, Spain, May 9-11, 2016*, pp. 884–892, 2016.

Lizotte, Daniel J., Wang, Tao, Bowling, Michael H., and Schuurmans, Dale. Automatic gait optimization with gaussian process regression. In *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, pp. 944–949, 2007.

Loshchilov, Ilya, Schoenauer, Marc, and Sebag, Michèle. Bi-population CMA-ES agorithms with surrogate models and line searches. In *Genetic and Evolutionary Computation Conference, GECCO '13, Amsterdam, The Netherlands, July 6-10, 2013, Companion Material Proceedings*, pp. 1177–1184, 2013.

Mannor, Shie, Rubinstein, Reuven, and Gat, Yohai. The Cross Entropy method for Fast Policy Search. In *Proceedings of the 20th International Conference on Machine Learning*, (ICML 2003), pp. 512–519, Washington, DC, USA, 2003.

Martinez-Cantin, Ruben. Bayesopt: A bayesian optimization library for nonlinear optimization, experimental design and bandits. *Journal of Machine Learning Research (JMLR)*, 15(1):3735–3739, January 2014. ISSN 1532-4435.

Mnih, Volodymyr, Kavukcuoglu, Koray, Silver, David, Rusu, Andrei A., Veness, Joel, Bellemare, Marc G., Graves, Alex, Riedmiller, Martin, Fidjeland, Andreas K., Ostrovski, Georg, Petersen, Stig, Beattie, Charles, Sadik, Amir, Antonoglou, Ioannis, King, Helen, Kumaran, Dharshan, Wierstra, Daan, Legg, Shane, and Hassabis, Demis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 02 2015.

Murray, Iain and Adams, Ryan P. Slice sampling covariance hyperparameters of latent gaussian models. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 1732–1740, 2010.

Peters, J., Mülling, K., and Altun, Y. Relative Entropy Policy Search. In *Proceedings of the 24th National Conference on Artificial Intelligence (AAAI)*. AAAI Press, 2010.

Russo, Daniel and Roy, Benjamin Van. Learning to optimize via posterior sampling. *Math. Oper. Res.*, 39(4): 1221–1243, 2014.

Schulman, John, Levine, Sergey, Jordan, Michael, and Abbeel, Pieter. Trust Region Policy Optimization. *International Conference on Machine Learning (ICML)*, pp. 16, 2015.

Shahriari, Bobak, Swersky, Kevin, Wang, Ziyu, Adams, Ryan P., and de Freitas, Nando. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.

Silver, David, Huang, Aja, Maddison, Chris J., Guez, Arthur, Sifre, Laurent, van den Driessche, George, Schrittwieser, Julian, Antonoglou, Ioannis, Panneershelvam, Veda, Lanctot, Marc, Dieleman, Sander, Grewe, Dominik, Nham, John, Kalchbrenner, Nal, Sutskever, Ilya, Lillicrap, Timothy, Leach, Madeleine, Kavukcuoglu, Koray, Graepel, Thore, and Hassabis, Demis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, January 2016.

Snoek, Jasper, Larochelle, Hugo, and Adams, Ryan P. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 2960–2968, 2012.

Snoek, Jasper, Swersky, Kevin, Zemel, Richard S., and Adams, Ryan P. Input warping for bayesian optimization of non-stationary functions. In *International Conference on Machine Learning (ICML)*, pp. 1674–1682, 2014.

Thomas, Philip S., Theocharous, Georgios, and Ghavamzadeh, Mohammad. High confidence policy improvement. In *International Conference on Machine Learning (ICML)*, pp. 2380–2388, 2015.

Vanhatalo, Jarno, Riihimäki, Jaakko, Hartikainen, Jouni, Jylänki, Pasi, Tolvanen, Ville, and Vehtari, Aki. Gpstuff: Bayesian modeling with gaussian processes. *Journal of Machine Learning Research (JMLR)*, 14(1):1175–1179, April 2013. ISSN 1532-4435.

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, Ziyu, Hutter, Frank, Zoghi, Masrour, Matheson, David, and de Freitas, Nando. Bayesian optimization in a billion dimensions via random embeddings. *J. Artif. Intell. Res. (JAIR)*, 55:361–387, 2016.

Wilson, Aaron, Fern, Alan, and Tadepalli, Prasad. Using trajectory data to improve bayesian optimization for reinforcement learning. *Journal of Machine Learning Research*, 15(1):253–282, 2014.