Unbounded Bayesian Optimization via Regularization

Bobak Shahriari
University of British Columbia
bshahr@cs.ubc.ca

Alexandre Bouchard-Côté
University of British Columbia

Nando de Freitas
University of Oxford
Google DeepMind

Abstract

Bayesian optimization has recently emerged as a powerful and flexible tool in machine learning for hyperparameter tuning and more generally for the efficient global optimization of expensive black box functions. The established practice requires a user-defined bounded domain, which is assumed to contain the global optimizer. However, when little is known about the probed objective function, it can be difficult to prescribe such a domain. In this work, we modify the standard Bayesian optimization framework in a principled way to allow for unconstrained exploration of the search space. We introduce a new alternative method and compare it to a volume doubling baseline on two common synthetic benchmarking test functions. Finally, we apply our proposed methods on the task of tuning the stochastic gradient descent optimizer for both a multi-layered perceptron and a convolutional neural network on the MNIST dataset.

1 Introduction

Since the technique was introduced over 50 years ago, Bayesian optimization has been applied to optimize black box objective functions in many different application domains. Perhaps the most relevant use-case in machine learning is the tuning of hyperparameters of computationally expensive models and algorithms [Bergstra et al., 2011, Mahendran et al., 2012, Snoek et al., 2012, Swersky et al., 2013, Hoffman et al., 2014]. However, the current state of the art requires the user to prescribe a bounded domain within which to search for the optimum. Unfortunately, setting these bounds—often done arbitrarily—is one of the main difficulties hindering the broader use of Bayesian optimization as a standard framework for hyperparameter tuning. For example, this obstacle was raised at the NIPS 2014 Workshop on Bayesian optimization as one of the open challenges in the field.

In the present work, we compare two methods that are capable of growing the search space as the optimization progresses. The first is a simple heuristic, based on an existing idea in optimization, which regularly doubles the volume of the search space throughout the procedure. Meanwhile, the second is a regularization method that is practical and easy to implement in any existing Bayesian optimization toolbox based on Gaussian Process priors over objective functions.

At a high level, our proposed regularization method is born out of the observation that the only component of Bayesian optimization that currently requires a bounding box on the search space is the maximization of the acquisition function. This constraint is necessary because acquisition functions can have suprema at infinity, so optimizing them may not return a point unless the feasible domain is constrained. By using a non-stationary prior mean as a regularizer, we can exclude this possibility and use an unconstrained optimizer, removing the need for a bounding box.

1.1 Related work

Although the notion of using a non-trivial Gaussian process prior mean is not new, it is usually expected to encode domain expert knowledge or known structure in the response surface. To the best of the authors' knowledge, only one recent work has considered using the prior mean as a regularization term and it was primarily to avoid selecting points along boundaries and in corners of the bounding box [Snoek et al., 2015].

In this work we demonstrate that a regularizing prior mean can be used to carry out Bayesian optimization without a rigid bounded domain. We compare this regularized approach to a *volume doubling* base-

Appearing in Proceedings of the 19^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2016, Cadiz, Spain. JMLR: W&CP volume 51.

line. While the regularized algorithms exhibit a much more homogeneous search behaviour (*i.e.* boundaries and corners are not disproportionately favoured), the volume doubling baseline performs very well in practice.

We begin with a brief review of Bayesian optimization with Gaussian processes in the next section, followed by an introduction to regularization via non-stationary prior means in Section 3, including visualizations that show that our proposed approach indeed ventures out of the initial user-defined bounding box. Section 4 reports our results on two synthetic benchmarking problems as well as two real hyperparameter tuning tasks, namely tuning the stochastic gradient descent optimizer of two neural network architectures on the MNIST handwritten digit recognition task.

2 Bayesian optimization

In this section, introduce some background on Bayesian optimization; see [Shahriari et al., 2016] for a more detailed review. Consider the problem of finding a global optimizer of an unknown objective function $f : \mathbb{R}^d \mapsto \mathbb{R}$, *i.e.* the problem of finding

$$\mathbf{x}_\star \in \arg\max_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}). \tag{1}$$

The function f is assumed to be a black-box for which we have no closed-form expression or available gradient information. We further assume that f is expensive to evaluate so we wish to locate the best possible input $\mathbf{x}$ with a relatively small budget of N evaluations. Finally, the evaluations $y \in \mathbb{R}$ of the objective function are noise-corrupted observations, and for the remainder of this work, we assume a Gaussian noise distribution, $y \mid \mathbf{x} \sim \mathcal{N}(f(\mathbf{x}), \sigma^2)$. Notice that, in contrast to typical Bayesian optimization settings, here we do not assume the arg max to be restricted to a bounded subset $\mathcal{X} \subset \mathbb{R}^d$.

Commonly known as *efficient global optimization*, this problem is poorly suited for popular non-convex methods such as Nelder–Mead and the more recent CMA-ES [Hansen and Ostermeier, 2001], because these require too many evaluations due to their inherently local search behaviour. In contrast, Bayesian optimization is a sequential model-based approach, which involves (i) maintaining a probabilistic *surrogate* model over likely functions given observed data; and (ii) sequentially selecting future query points according to a selection *policy*, which leverages the uncertainty in the surrogate to negotiate exploration of the search space and exploitation of currently suspected modes. The selection policy is represented by an *acquisition function* $\alpha_n : \mathbb{R}^d \mapsto \mathbb{R}$, where the subscript indicates the implicit dependence on the surrogate and, by extension, on the observed data $\mathcal{D}_n = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$.

More precisely, at iteration n: an input $\mathbf{x}_{n+1}$ is selected by maximizing the acquisition function α_n; the black-box is queried and produces a noisy y_{n+1}; and the surrogate is updated in light of the new data point $(\mathbf{x}_{n+1}, y_{n+1})$. Finally, after N queries the algorithm must make a final recommendation $\hat{\mathbf{x}}_N$ which represents its best estimate of the optimizer.

Algorithm 1 Bayesian optimization framework with best-observation recommendation

```
initialize prior surrogate f̂_0
for n = 1, ..., N do
    derive index α_{n−1} from surrogate f̂_{n−1}
    select next x_n ∈ arg max_{x∈X} α_{n−1}(x)
    query black-box y_n ~ N(f(x_n), σ²)
    update surrogate f̂_n given previous surrogate
    f̂_{n−1} and new observation (x_n, y_n)
end for
choose j ∈ arg max_{i=1:N} y_i
return x̂_N^obs = x_j
```

Recommendation strategy. In our experiments, we exclusively use the *best-observation* recommendation strategy, which returns $\hat{\mathbf{x}}_N^{\text{obs}} = \mathbf{x}_j$ where $j \in \arg\max_{i=1:N} y_i$. This is a particularly convenient choice when applying model-based optimization to real tasks where the surrogate model is likely to be misspecified—as in our neural network tuning tasks. Intuitively, the surrogate is trusted to make selections throughout the exploration phase, while only the observed data is trusted when making a final recommendation. Alternatively, in the presence of large noise, one could recommend the *incumbent*, which is similar to best-observation but uses the prediction at observed inputs instead of the observed y_i, *i.e.* $\hat{\mathbf{x}}_N^{\text{inc}} = \mathbf{x}_j$ where $j \in \arg\max_{i=1:N} \hat{f}_N(\mathbf{x}_i)$ and $\hat{f}_N$ is the surrogate after N observations. Finally, the best-latent strategy recommends a point $\hat{\mathbf{x}}_N^{\text{lat}} \in \arg\max_{\mathbf{x}} \hat{f}_N(\mathbf{x})$ where $\hat{f}(\mathbf{x})$ is our surrogate model prediction at $\mathbf{x}$.

2.1 Gaussian processes

Perhaps the most common surrogate in Bayesian optimization, and the one we prescribe in this work, is the Gaussian process (GP) prior over functions [Rasmussen and Williams, 2006]. When combined with a Gaussian likelihood, the posterior is also a GP and the Bayesian update of the surrogate can be computed analytically. Under the assumption that the black-box objective function f is sampled from a GP prior, two Bayesian optimization acquisition functions have recently been proven

to converge, when using Gaussian process surrogates; namely GP-UCB [Srinivas et al., 2010] and expected improvement [Bull, 2011]. Note that random forests have also been proposed in the model-based optimization literature [Hutter et al., 2010].

A Gaussian process $\mathrm{GP}(\mu_0, k)$ is fully characterized by its prior mean function $\mu_0 : \mathbb{R}^d \mapsto \mathbb{R}$ and its positive-definite kernel, or covariance function, $k : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$. Given any finite collection[1] of n points $\mathbf{x}_{1:n}$, the values of $f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)$ are jointly Gaussian with mean $\mathbf{m}$, where $m_i = \mu_0(\mathbf{x}_i)$, and $n \times n$ covariance matrix $\mathbf{K}$, where $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$—hence the term covariance function.

Given the Gaussian likelihood model, the vector of concatenated observations $\mathbf{y} = y_{1:n}$ is also jointly Gaussian with covariance $\mathbf{K} + \sigma^2\mathbf{I}$. Therefore, at any arbitrary test location $\mathbf{x}$, we can query our surrogate model (the GP) for the predicted function value $\hat{f}_n(\mathbf{x})$ conditioned on observed data $\mathcal{D}_n$. By straightforward multivariate Gaussian conditioning, the quantity $\hat{f}_n(\mathbf{x})$ is a Gaussian random variable with the following mean $\mu_n(\mathbf{x})$ and marginal variance $\sigma_n^2(\mathbf{x})$

$$\mu_n(\mathbf{x}) = \mu_0(\mathbf{x}) + \mathbf{k}(\mathbf{x})^\mathsf{T}(\mathbf{K} + \sigma^2\mathbf{I})^{-1}(\mathbf{y} - \mathbf{m}), \quad (2)$$

$$\sigma_n^2(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}) - \mathbf{k}(\mathbf{x})^\mathsf{T}(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{k}(\mathbf{x}), \quad (3)$$

where $\mathbf{k}(\mathbf{x})$ is the vector of cross-covariance terms between the test point $\mathbf{x}$ and the observed data $\mathbf{x}_{1:n}$.

There are a plethora of positive-definite kernels that can be combined by sums and products to create new such kernels with rich structure, However, throughout this work we use the following simple squared exponential kernel, noting that our proposed method is readily extended to other kernels:

$$k_{\text{SE}}(\mathbf{x}, \mathbf{x}') = \theta_0 \exp(-\tfrac{1}{2}r^2), \quad (4)$$

where $r = (\mathbf{x} - \mathbf{x}')^\mathsf{T}\mathbf{\Lambda}^{-1}(\mathbf{x} - \mathbf{x}')$ and $\mathbf{\Lambda}$ is a diagonal matrix of d length scales $\theta_{1:d}$ and θ_0 is the kernel amplitude. Collectively referred to as the *hyperparemeter*, the vector $\boldsymbol{\theta} = \theta_{0:d}$ parameterizes the kernel function k. When the noise variance σ^2 is unknown, it can be added as a model hyperparameter as well. Similarly, the most common agnostic choice of prior mean is a constant bias $\mu_0(\mathbf{x}) \equiv b$, which, with a slight abuse of notation, we add to the vector $\boldsymbol{\theta}$.

Hyperparameter marginalization. As in many regression tasks, the hyperparameter $\boldsymbol{\theta}$ must somehow be specified and has a dramatic effect on performance. Common tuning techniques such as cross-validation and maximum likelihood are either highly data-inefficient or run the risk of overfitting. Recently, a Bayesian treatment of the hyperparameters via Markov chain Monte Carlo (MCMC) has become standard practice in Bayesian optimization [Snoek et al., 2012]. Similarly in the present work, we specify an uninformative prior on $\boldsymbol{\theta}$ and approximately integrate it by sampling from the posterior $p(\boldsymbol{\theta}|\mathcal{D}_n)$ via slice sampling.

[1]Here we use the convention $a_{i:j} = \{a_i, \ldots, a_j\}$.

2.2 Acquisition functions

So far we have described the probabilistic model we use to represent our prior belief about the unknown objective f, and how to update this belief given observations $\mathcal{D}_n$ with (2) and (3). We have not described any mechanism or policy for selecting the sequence of query points $\mathbf{x}_{1:n}$. One could select these arbitrarily or by grid search but, in the spirit of data-efficient optimization, this would be wasteful—uniformly random search has also been proposed as a surprisingly good alternative to grid search [Bergstra and Bengio, 2012]. There is, however, a rich literature on selection strategies that utilize the surrogate model to guide the sequential search, *i.e.* the selection of the next query point $\mathbf{x}_{n+1}$ given $\mathcal{D}_n$.

The key idea behind these strategies is to define acquisition functions $\alpha_n : \mathbb{R}^d \mapsto \mathbb{R}$, which quantify the promise[2] of any point in the search space. The acquisition function is carefully designed to trade off exploration of the search space and exploitation of promising neighborhoods, given the surrogate model $\hat{f}_n$. There are three common types of acquisition functions: improvement-based, optimistic, and information-based policies.

The improvement-based acquisition functions, probability and expected improvement (PI and EI, respectively), select the next point with the most probable and most expected improvement, respectively [Kushner, 1964, Močkus et al., 1978]. On the other hand, the optimistic policy *upper confidence bound* (GP-UCB) measures, marginally for each test point $\mathbf{x}$, how good the corresponding observation y will be in a low and fixed probability "good case scenario"—hence the optimism [Srinivas et al., 2010]. In contrast, there exist information-based methods such as randomized probability matching, also known as Thompson sampling [Thompson, 1933, Scott, 2010, Chapelle and Li, 2011, Kaufmann et al., 2012, Agrawal and Goyal, 2013], or the more recent *entropy search* methods [Villemonteix et al., 2009, Hennig and Schuler, 2012, Hernández-Lobato et al., 2014]. Thompson sampling selects the next point

[2]The way "promise" is quantified depends on whether we care about cumulative losses of the intermediate selections $\mathbf{x}_1 : N$ or only the loss of the final recommendation $\hat{\mathbf{x}}_N$.

according to the distribution of the optimum $\mathbf{x}_\star$, which is induced by the current posterior [Scott, 2010, Hernández-Lobato et al., 2014, Shahriari et al., 2014]. Meanwhile, entropy search methods select the point $\mathbf{x}$ that is expected to provide the most information towards reducing uncertainty about $\mathbf{x}_\star$.

Expected improvement. In this work, we focus our attention on EI, which is perhaps the most common acquisition function [Jones et al., 1998, Jones, 2001]. Using a Gaussian process surrogate model, the expected improvement upon a fixed *target* τ can be computed analytically, yielding the following expression

$$\alpha_n^{\mathrm{EI}}(\mathbf{x}) = (\mu_n(\mathbf{x}) - \tau)\Phi(z) + \sigma_n(\mathbf{x})\phi(z), \qquad (5)$$

where $z = \frac{\mu_n(\mathbf{x})-\tau}{\sigma_n(\mathbf{x})}$, and Φ and ϕ denote the standard normal cumulative distribution and density functions, respectively. Note however that the technique we outline in the next section can readily be extended to any Gaussian process derived acquisition function, including all those mentioned above.

3 Unbounded Bayesian optimization

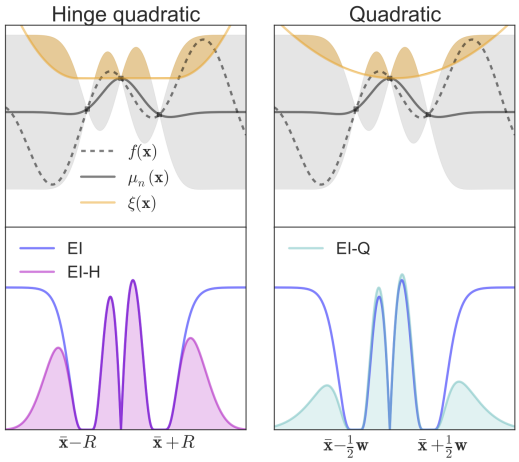
In this section, we introduce two methods that will result in robustness to the choice of initial bounding box. The first is a simple approach we call volume doubling and the second is out proposed approach, which can be interpreted as a regularization of current methods.

3.1 Volume doubling

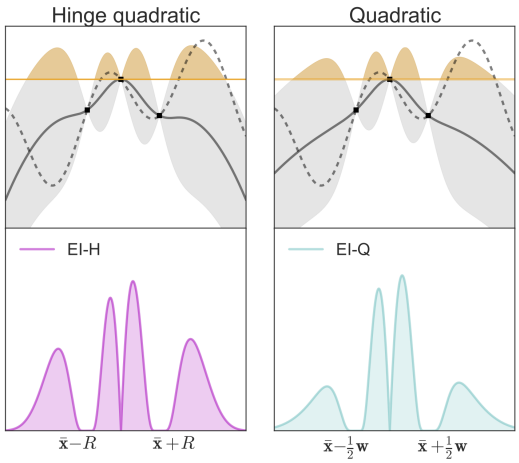
This heuristic consists of expanding the search space regularly as the optimization progresses, starting with an initial user-defined bounding box. This method otherwise follows the standard Bayesian optimization procedure and optimizes within the bounding box that is available at the given time step n. This approach requires two parameters: the number of iterations between expansions and the growth factor γ. Naturally, to avoid growing the feasible space $\mathcal{X}$ by a factor that is exponential in d, the growth factor applies to the volume of $\mathcal{X}$. Finally, the expansion is isotropic about the centre of the domain. In this work, we double ($\gamma = 2$) the volume every $3d$ evaluations (only *after* an initial latin hypercube sampling of $3d$ points).

3.2 Regularizing improvement policies

We motivate the regularized approach by considering improvement policies, in particular, EI. However, in the next section we show that this proposed approach can be extended more generally to all GP-derived acquisition functions, and in fact it is not difficult to apply the idea to other surrogate models, which we leave for future work.



(a) Minimum improvement view



(b) Prior mean view

Figure 1: Visualization of the two alternate views of regularization in Bayesian optimization. The objective function and posterior surrogate mean are represented as black dashed and solid lines, respectively, with grey shaded areas indicating $\pm 2\sigma_n$. Integrating the surrogate model above the target (orange shaded area) results in the regularized EI acquisition function (magenta and cyan). Using a non-stationary target with a constant prior mean (left) or a fixed target with a non-stationary prior mean (right) lead to indistinguishable acquisition functions, which decay at infinity.

Improvement policies are a popular class of acquisition functions that rely on the improvement function

$$I(\mathbf{x}) = (f(\mathbf{x}) - \tau)\mathbb{I}[f(\mathbf{x}) \geq \tau] \qquad (6)$$

where τ is some target value to improve upon. Expected improvement then compute $\mathbb{E}[I(\mathbf{x})]$ under the posterior GP.

When the optimal objective value $f_\star$ is known and we set $\tau = f_\star$, these algorithms are referred to as *goal seeking* [Jones, 2001]. When the optimum is not known, it is common to use a proxy for $f_\star$, such as the value of the best observation so far, y^+; or in the noisy setting, one can use either the maximum of the posterior mean or the value of the mean prediction μ_n at the *incumbent* $\mathbf{x}^+ \in \arg\max_{\mathbf{x}\in\mathbf{x}_{1:n}} \mu_n(\mathbf{x})$.

In some cases, the above choice of target $\tau = y^+$ can lead to a lack of exploration, therefore it is common to choose a *minimum improvement* parameter $\xi > 0$ such that $\tau = (1+\xi)y^+$ (for convenience here we assume y^+ is positive and in practice one can subtract the overall mean to make it so). Intuitively, the parameter ξ allows us to require a minimum fractional improvement over the current best observation y^+. Previously, the parameter ξ had always been chosen to be constant,[3] if not zero. In this work we propose to use a function $\xi : \mathbb{R}^d \mapsto \mathbb{R}^+$ which maps points in the space to a value of fractional minimum improvement. Following the same intuition, the function ξ lets us require larger improvements from points that are *farther* and hence acts as a regularizer that penalizes distant points. The improvement function hence becomes:

$$I(\mathbf{x}) = (f(\mathbf{x}) - \tau(\mathbf{x}))\mathbb{I}[f(\mathbf{x}) \geq \tau(\mathbf{x})], \tag{7}$$

where the target is now a function of $\mathbf{x}$:

$$\tau(\mathbf{x}) = (1 + \xi(\mathbf{x}))y^+; \tag{8}$$

the choice of $\xi(\mathbf{x})$ is discussed in the Section 3.2.2.

3.2.1 Extension to general policies

In the formulation of the previous section, our method seems restricted to improvement policies. However, many recent acquisition functions of interest are not improvement-based, such as GP-UCB, entropy search, and Thompson sampling. In this section, we describe a closely related formulation that generalizes to all acquisition functions that are derived from a GP surrogate model.

Consider expanding our choice of non-stationary target τ in Equation (8)

$$\begin{aligned} I(\mathbf{x}) &= (f(\mathbf{x}) - y^+(1 + \xi(\mathbf{x})))\mathbb{I}[f(\mathbf{x}) \geq y^+(1 + \xi(\mathbf{x}))] \\ &= (f(\mathbf{x}) - y^+\xi(\mathbf{x}) - y^+)\mathbb{I}[f(\mathbf{x}) - y^+\xi(\mathbf{x}) \geq y^+] \\ &= (\tilde{f}(\mathbf{x}) - y^+)\mathbb{I}[\tilde{f}(\mathbf{x}) \geq y^+] \end{aligned} \tag{9}$$

where $\tilde{f}$ is the posterior mean of a GP from (2) with prior mean $\tilde{\mu}_0(\mathbf{x}) = \mu_0(\mathbf{x}) - y^+\xi(\mathbf{x})$. Notice the similarity between (6) and (9). Indeed, in its current form we see that the regularization can be achieved simply by using a different prior mean $\tilde{\mu}_0$ and a constant target y^+. This duality can be visualized when comparing the left and right panel of Figure 1.

Strictly speaking, Equations (6) and (9) are not exactly equivalent. Indeed, using a surrogate GP with prior mean $\tilde{\mu}_0$, the posterior mean yields an additional term

$$-\mathbf{k}(\mathbf{x})^\mathsf{T}(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\xi(\mathbf{X}), \tag{10}$$

where $[\xi(\mathbf{X})]_i = \xi(\mathbf{x}_i)$. This negative term will only accentuate the vanishing of expected improvement for test points $\mathbf{x}$ that are far from the regularizer centre. Indeed, in Figure 1 the two views produce indistinguishable regularized acquisition functions (in this case EI). However, we favour this new formulation because we can apply the same regularization to any policy which uses a Gaussian process, namely Thompson sampling and entropy search.

3.2.2 Choice of regularization

By inspecting Equation (5), we see that any *coercive* prior mean function would lead to an asymptotically vanishing EI acquisition function as $||\mathbf{x}|| \to \infty$. More precisely, this is due to both Φ and ϕ vanishing as their arguments approach $-\infty$. In this work, we consider two coercive regularizing prior mean functions, namely a quadratic (Q) and an isotropic hinge-quadratic (H), defined as follows (excluding the constant bias b)

$$\xi_Q(\mathbf{x}) = (\mathbf{x} - \bar{\mathbf{x}})^\mathsf{T}\mathrm{diag}(\mathbf{w}^2)^{-1}(\mathbf{x} - \bar{\mathbf{x}}), \tag{11}$$

$$\xi_H(\mathbf{x}) = \mathbb{I}[\|\mathbf{x} - \bar{\mathbf{x}}\|_2 > R]\left(\frac{\|\mathbf{x} - \bar{\mathbf{x}}\|_2 - R}{\beta R}\right)^2. \tag{12}$$

Both of these regularizers are parameterized by d location parameters $\bar{\mathbf{x}}$, and while ξ_Q has an additional d width parameters $\mathbf{w}$, the isotropic regularizer ξ_H has a single radius R and a single β parameter, which controls the curvature of ξ_H outside the ball of radius R; in what follows we fix $\beta = 1$.

Fixed prior mean hyperparameters. We are left with the choice of centre $\bar{\mathbf{x}}$ and radius parameters R (or widths $\mathbf{w}$). Unlike the bias term b, these parameters of the regularizer are not intended to allow the surrogate to better fit the observations $\mathcal{D}_n$. In fact, using the marginal likelihood to estimate or marginalize $\psi = \{\bar{\mathbf{x}}, R, \mathbf{w}\}$, could lead to fitting the regularizer to a local mode which could trap the algorithm in a suboptimal well. For this reason, we use an initial, temporary user-defined bounding box to set ψ at the beginning of the run; the value of ψ remains fixed in all subsequent iterations.

Note that, while the bounding box in current Bayesian optimization practice is a hard constraint on the do-

[3] Here we mean constant with respect to $\mathbf{x}$, there has been previous work on adaptively scheduling this parameters.

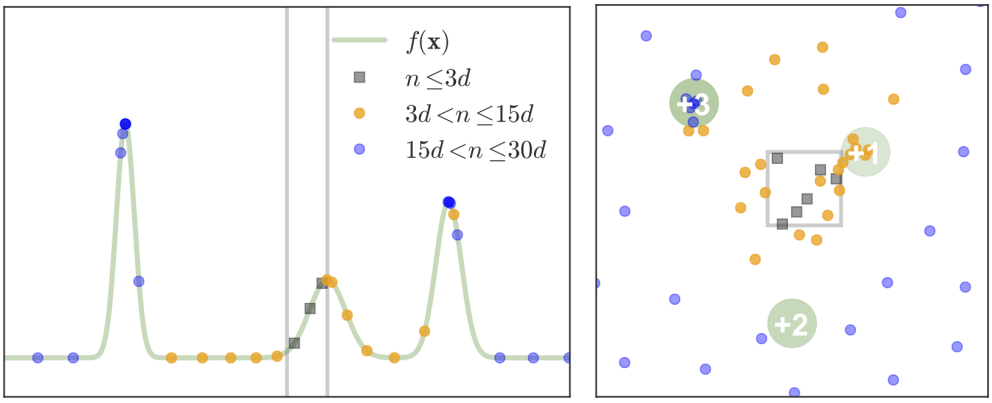


Figure 2: Visualization of selections $\mathbf{x}_n$ made by the EI-H algorithm on two toy problems: three Gaussian modes in one (left) and two (right) dimensions. Grey lines delimit the initial bounding box; grey square markers indicate the initial latin hypercube points, while the orange and blue points distinguish between the first and second half of the evaluation budget of $30d$, respectively. In the two-dimensional example, the height of the Gaussians are indicated by $+1$, $+2$, and $+3$.

main, our approach uses it simply to generate a regularizer which will focus the sequential search without constraining it. One clear advantage of our algorithm is that it adheres to the current user interface, allowing practitioners to continue simply specifying a reasonable range for each search dimension. This is arguably a much more natural requirement than a multidimensional centre and width parameters.

Finally, note that when users specify an arbitrary bounding box, they are in effect fixing $2d$ parameters. In that respect, our algorithm requires no more parameters than current practice, yet allows the search to progress outside of this arbitrary box, given a large enough budget.

3.3 Visualization

Before describing our experimental results, Figure 2 provides a visualization of EI with the hinge-quadratic prior mean, optimizing two toy problems in one and two dimensions. The objective functions simply consist of three distant Gaussian modes of varying heights and the initial bounding box is set such that it does not include the optimum. We draw attention to the way the space is gradually explored outward from the initial bounding box.

4 Experiments

In this section, we evaluate our proposed methods and show that they achieve the desirable behaviour on two synthetic benchmarking functions, and a simple task of tuning the stochastic gradient descent and regularization parameters used in training a multi-layered perceptron (MLP) and a convolutional neural network (CNN) on the MNIST dataset.

Experimental protocol. For every test problem of dimension d and every algorithm, the optimization was run with an overall evaluation budget of $30d$ including an initial $3d$ points sampled according to a latin hypercube sampling scheme (as suggested in [Jones, 2001]). Throughout each particular run, at every iteration n we record the value of the best observation up to n and report these in Figure 3. Experiments were repeated to report and compare the mean and standard error of the algorithms: the synthetic experiments were repeated 40 times, while the MNIST experiments were repeated 25 and 20 times for the MLP and the CNN, respectively.

Algorithms. We compared the two different methods from Section 3 to the standard EI with a fixed bounding box. Common random seeds were used for all methods in order to reduce confounding noise. All algorithms were implemented in the `pybo` framework available on github,[4] and are labelled in the following figures as follows:

EI: Vanilla expected improvement with hyperparameter marginalization via MCMC.

EI-V: Expected improvement with the search volume doubled every $3d$ iterations.

EI-H: Regularized EI with a hinge-quadratic prior mean with $\beta = 1$ and R fixed by the circumradius of the initial bounding box.

[4] https://github.com/mwhoffman/pybo

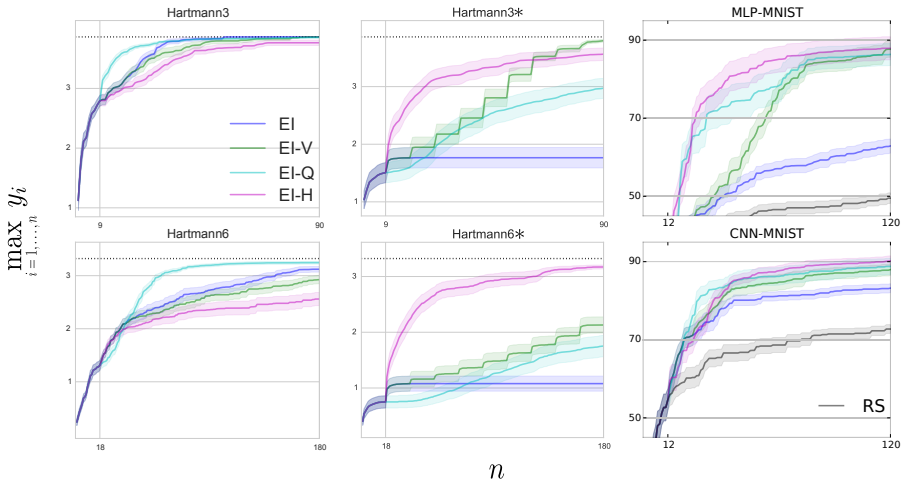


Figure 3: Best observations as optimization progresses. In the Hartmann experiments, the known optimum is represented as a horizontal dotted line. Plotting mean and standard error over 40 (Hartmann), 25 (MLP), and 20 (CNN) repetitions.

EI-Q: Regularized EI with a quadratic prior mean where the widths $\mathbf{w}$ are fixed to those of the initial bounding box.

RS: As an additional benchmark, on the neural network tuning tasks, we considered a random selection strategy, which uniformly sampled within the user-defined bounding box.

Note that for the regularized methods EI-H/Q, the initial bounding box is only used to fix the location and scale of the regularizers, and to sample initial query points. In particular, both regularizers are centred around the box centre. For the quadratic regularizer the width of the box in each direction is used to fix $\mathbf{w}$, whereas for the hinge-quadratic R is set to the box circumradius. Once these parameters are fixed, the bounding box is no longer relevant, and more importantly, the algorithm is free to select points outside of it, see Figure 2 and 4 for example.

4.1 Synthetic benchmarks: Hartmann

The Hartmann 3 and 6 functions (numbers refer to their dimensionality) are standard, synthetic global optimization benchmarking test functions with known global optima. These are typically optimized in the unit hypercube $[0, 1]^d$, as we do in our Hartmann3 and 6 experiments.

In a separate experiment, indicated by an appended asterisk (*e.g.* Hartmann3$*$), we consider an initial bounding box of side length 0.2 centred uniformly at random within the unit hypercube. Each of the 40 repetitions of this experiment fixed a different such domain for all algorithms. The smaller domain has a 0.2^d probability of including the global optimum, especially unlikely in the six-dimensional problem. This experiment is meant to test whether our proposed methods are capable of useful exploration outside the initial bounding box and further compare them in such a situation.

4.2 MLP and CNN on MNIST

The MNIST hand-written digit recognition dataset is a very common task for testing neural network methods and architectures. Neural networks are usually trained using some variant of stochastic gradient descent (SGD). The hyperparameters can impact both the speed of convergence and the quality of the trained network. We consider an MLP with 2048 hidden units with tanh non-linearities, and a CNN with two convolutional layers. These examples were taken from the official GitHub repository of `torch` demos.[5] The code written for this work can be readily extended to any other demo in the repository or in fact any script that can be run from the shell.

In this experiment, we optimize four parameters of the SGD optimizer, namely the learning rate and momentum, and the ℓ_1 and ℓ_2 regularization coefficients. The parameters were optimized in log space (base e) with an initial bounding box of $[-3, -1] \times [-3, -1] \times [-3, 1] \times [-3, 1]$, respectively. For each parameter setting, a black-box function evaluation corresponds to either training the MLP for 5 epochs or the CNN for 3, and returning the test set accuracy. To be clear, the goal of this experiment is not to achieve state-of-the-art for this classification task but instead to demon-

[5] https://github.com/torch/demos

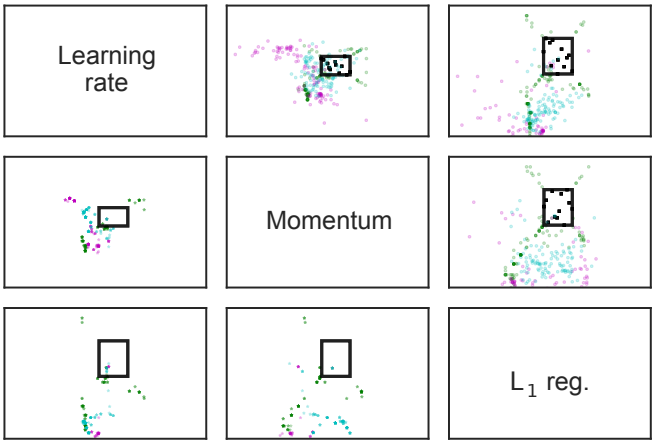


Figure 4: Pairwise scatterplot of selections (upper triangle) and recommendations (lower triangle) for the MLP-MNIST experiment. For example, the second plot of the first row corresponds to a scatter plot of the selected learning rates vs. momentum parameters *for a single seed*. In contrast, the first plot of the second row corresponds to a scatter plot of the recommended learning rates and momentum parameters *over all runs*. The initial bounding box and sample points (for this particular run) are shown as a black rectangle and black square dots, respectively. All other points respect the color scheme of Figure 3. (the ℓ_2 regularization parameters were cropped out for space considerations.)

strate that our proposed algorithms can find optima well outside their initial bounding boxes.

4.3 Results

Figure 3 shows that for the Hartmann tests, the proposed Bayesian optimization approaches work well in practice. The results confirm our hypothesis that the proposed methods are capable of useful exploration outside the initial bounding box. We note that when using the entire unit hypercube as the initial box, all the Bayesian optimization techniques exhibit similar performance as in this case the optimum is within the box. The Hartmann tests also show that the volume doubling heuristic is a good baseline method; and the plateaus suggest that this method warrants further study in, perhaps adaptive, scheduling strategies. Although it is less effective than EI-H as the dimensionality increases, it is nonetheless an improvement over standard EI in all cases.

The MNIST experiment shows good performance from all three methods EI-{V,H,Q}, particularly from the hinge-quadratic regularized algorithm. Indeed, when compared to the standard EI, EI-H boasts over 20% improvement in accuracy on the MLP and almost 10% on the CNN.

We believe that EI-H performs particularly well in settings where a small initial bounding box is prescribed because the hinge-quadratic regularizer allows the algorithm to explore outward more quickly. In contrast, EI-Q performs better when the optimum is included in the initial box; we suspect that this is due to the fact that the regularizer avoids selecting boundary and corner points, which EI and EI-V tend to do, as can be seen in Figure 4.

Figure 4 demonstrates that the algorithm in fact explores points outside the initially suggested bounded domain (drawn as a black rectangle). Indeed, while the green dots (EI-V) follow the corners of the growing bounding box, the magenta and cyan dots of EI-H/Q, respectively, do not exhibit this artefact.

5 Conclusion and future work

In this work, we propose a versatile new approach to Bayesian optimization which is not limited to a search within a bounding box. Indeed, given an initial bounding box that does not include the optimum, we have demonstrated that our approach can expand its region of interest and achieve greater function values. Our method fits seamlessly within the current Bayesian optimization framework, and can be readily used with any acquisition function which is induced by a GP.

We emphasize that in this work we have addressed one of the challenges that must be overcome toward the development of a practical Bayesian optimization tool for hyper-parameter tuning and efficient global optimization in general. A complete solution, however, must also address the issues of dimensionality, non-stationarity, and early stopping.

References

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

J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13:281–305, 2012.

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

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

O. Chapelle and L. Li. An empirical evaluation of Thompson sampling. In *Advances in Neural Information Processing Systems*, pages 2249–2257, 2011.

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

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

J. M. Hernández-Lobato, M. W. Hoffman, and Z. Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. In *Advances in Neural Information Processing Systems*. 2014.

M. W. Hoffman, B. Shahriari, and N. de Freitas. On correlation and budget constraints in model-based bandit optimization with application to automatic machine learning. In *AI and Statistics*, pages 365–374, 2014.

F. Hutter, T. Bartz-Beielstein, H. H. Hoos, K. Leyton-Brown, and K. P. Murphy. Sequential model-based parameter optimisation: an experimental investigation of automated and interactive approaches. In T. Bartz-Beielstein, M. Chiarandini, L. Paquete, and M. Preuss, editors, *Empirical Methods for the Analysis of Optimization Algorithms*, chapter 15, pages 361–411. Springer, 2010.

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

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

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

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

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

J. Močkus, V. Tiesis, and A. Žilinskas. The application of bayesian methods for seeking the extremum. In L. Dixon and G. Szego, editors, *Toward Global Optimization*, volume 2. Elsevier, 1978.

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

S. L. Scott. A modern Bayesian look at the multi-armed bandit. *Applied Stochastic Models in Business and Industry*, 26(6):639–658, 2010.

B. Shahriari, Z. Wang, M. W. Hoffman, A. Bouchard-Côté, and N. de Freitas. An entropy search portfolio. In *NIPS workshop on Bayesian Optimization*, 2014.

B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.

J. Snoek, H. Larochelle, and R. P. Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, pages 2951–2959, 2012.

J. Snoek, O. Rippel, K. Swersky, R. Kiros, N. Satish, N. Sundaram, M. Patwary, M. Ali, R. P. Adams, et al. Scalable Bayesian optimization using deep neural networks. *International Conference on Machine Learning*, 2015.

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

K. Swersky, J. Snoek, and R. P. Adams. Multi-task Bayesian optimization. In *Advances in Neural Information Processing Systems*, pages 2004–2012, 2013.

W. R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

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