An Entropy Search Portfolio for Bayesian Optimization

Bobak Shahriari\(^1\) bshahr@cs.ubc.ca
Ziyu Wang\(^2\) ziyu.wang@cs.ox.ac.uk
Matthew W. Hoffman\(^3\) mwh30@cam.ac.uk
Alexandre Bouchard-Côté\(^4\) bouchard@stat.ubc.ca
Nando de Freitas\(^2,4\) nando@cs.ox.ac.uk

\(^1\)University of British Columbia, Canada
\(^2\)University of Oxford, United Kingdom
\(^3\)University of Cambridge, United Kingdom
\(^4\)Canadian Institute for Advanced Research

Abstract

Portfolio methods provide an effective, principled way of combining a collection of acquisition functions in the context of Bayesian optimization. We introduce a novel approach to this problem motivated by an information theoretic consideration. Our construction additionally provides an extension of Thompson sampling to continuous domains with GP priors. We show that our method outperforms a range of other portfolio methods on several synthetic problems, automated machine learning tasks, and a simulated control task. Finally, the effectiveness of even the random portfolio strategy suggests that portfolios in general should play a more pivotal role in Bayesian optimization.

1 Introduction

Bayesian optimization is a popular and very successful set of techniques for global optimization of black-box functions that are costly to evaluate. Such methods address the problem of finding the minimizer of a nonlinear function \( f \) which is generally non-convex and multi-modal, and whose derivatives are often unavailable. Major applications of these techniques include interactive user-interfaces \([6, 4]\), robotics \([18, 21]\), environmental monitoring \([20]\), information extraction \([33]\), combinatorial optimization \([16, 34]\), sensor networks \([10, 28]\), adaptive Monte Carlo \([19, 32]\), experimental design \([1]\), and reinforcement learning \([5]\).

While this approach is very widely applicable—it can be used anywhere one can evaluate the target function at arbitrary points, even subject to observation noise—one area of particular interest is the automatic tuning of machine learning algorithms \([2, 27, 29, 30, 15]\). In this setting the optimized quantity often corresponds to the accuracy of a learning algorithm as a function of different algorithmic choices or hyperparameters. Whereas these parameters have traditionally been selected by hand, this approach is quickly becoming unwieldy, if not impossible, with the size and complexity of modern statistical models.

The key idea underlying Bayesian optimization is that one can place a prior over the space of probable functions before optimization. This prior can then be sequentially combined with data as it is observed in order to form an updated posterior over likely functions. In particular, the point selected by the algorithm at each iteration will explicitly depend on the posterior. While this general principle encompasses all of Bayesian optimization, naturally, how well the algorithm performs, i.e. how quickly it is able to find the global maximizer in practice, hinges on exactly how the posterior...
is utilized. We will refer to the method which maps observed data to a potential query point as an acquisition strategy.

One early acquisition strategy utilized in the literature, Probability of Improvement (PI), selects the candidate point which has maximum probability of improving over the best point seen so far (the incumbent). The point with the highest probability of improvement, however, frequently lies close to the current incumbent. As a result, in practice the use of PI often exhibits very greedy behavior. Instead, one can take into account the expected improvement, resulting in the Expected Improvement (EI) strategy [23]. More recently a wide variety of more advanced techniques have been proposed which include Bayesian upper confidence bounds [28] and information-theoretic approaches [31] [11]. However, no single acquisition strategy provides better performance over all problem instances. In fact, a frequent occurrence is that the optimality of various strategies can swap during the course of optimization. To remedy this, Hoffman et al. propose the use of a portfolio containing multiple acquisition strategies [13]. A key ingredient of this approach is the meta-criterion by which a portfolio method selects among different strategies. This function is analogous to an acquisition function, but at a higher level. Whereas acquisition functions assign utility to points in the input space, a meta-criterion assigns utility to candidates suggested by base strategies.

Our contribution in this work is the proposal of an information-theoretic meta-criterion similar to that used by [31] [11], which we call the Entropy Search Portfolio (ESP). Information-theoretic methods aim to reduce the entropy of the location of the minimum of the black-box function. Estimating the probability of a point being the minimizer as well as estimating the entropy of this probability distribution is not straightforward and often requires discretization of the search space. Hennig et al. do this by slice sampling from an acquisition function such as PI or EI [11]. In practice, we have found this process to be inefficient and numerically unstable. To improve on this, we introduce a novel way of obtaining discrete search points by sampling from the probability distribution over minimizers approximately. We do so by using a kernel approximation originally proposed by Rahimi et al. [25]. Using this technique, we are also able to draw and optimize samples of a Gaussian process over a continuous space \( \mathcal{X} \) approximately. This gives us an important extension of Thompson sampling, a popular bandit strategy that has so far been confined to the multi-armed bandit setting (discrete space \( \mathcal{X} \) ) [7] [15]. We also include this new method as an additional acquisition strategy in all portfolios that we considered in this work. Finally, we provide empirical evidence that our approach results in gains not only over previous portfolio strategies but also over the fundamental strategies that make up the portfolio.

2 Entropy search over portfolios

As described in the previous section, we are interested in finding the global minimizer \( \mathbf{x}_* = \arg\min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}) \) of a function \( f \) over some bounded domain, typically \( \mathcal{X} \subset \mathbb{R}^d \). We further assume that \( f(\mathbf{x}) \) can only be evaluated via queries to a black-box that provides noisy outputs from some set, typically \( \mathcal{Y} = \mathbb{R} \). For this work we assume \( y_t \sim \mathcal{N}(f(\mathbf{x}_t), \sigma^2) \), however, our framework can be extended to other non-Gaussian likelihoods. In this setting, we describe a sequential search algorithm that, after \( t \) iterations, proposes to evaluate \( f \) at some location \( \mathbf{x}_{t+1} \) given by an acquisition strategy \( \alpha(D_t) \) where \( D_t = \{(x_1, y_1), \ldots, (x_t, y_t)\} \) is the history of previous observations. More formally an acquisition strategy takes the form of a function \( \alpha : \mathcal{H} \to \mathcal{X} \) where \( \mathcal{H} = \bigcup_{j} (\mathcal{X} \times \mathcal{Y})^j \) is the set of all possible histories. Finally, after \( T \) iterations the algorithm must make a final recommendation \( \mathbf{x}_T \), i.e. its best estimate for the optimum.

The particular set of strategies we focus on take a Bayesian approach to modeling the unobserved function \( f \) and utilize a posterior distribution over this function to guide the search process. In this work we use a zero-mean Gaussian process (GP) prior for \( f \) [26]. This prior is specified by a positive-definite kernel function \( k(\mathbf{x}, \mathbf{x}') \). Given any finite collection of points \( \{x_1, \ldots, x_t\} \), the values of \( f \) at these points are jointly zero-mean Gaussian with covariance matrix \( \mathbf{K}_t \), where \( [\mathbf{K}_{t}]_{ij} = k(x_i, x_j) \). For the Gaussian likelihood described above, the vector of concatenated observations \( \mathbf{y}_t \) is also jointly Gaussian with zero-mean. Therefore, at any location \( \mathbf{x} \), the latent function \( f(\mathbf{x}) \) conditioned on \( D_t \) is Gaussian with marginal mean \( \mu_{t}(\mathbf{x}) \) and variance \( \sigma_t^2(\mathbf{x}) \) given by

\[
\mu_t(\mathbf{x}) = \mathbf{k}_t(\mathbf{x})^T(\mathbf{K}_t + \sigma^2\mathbf{I})^{-1}\mathbf{y}_t, \quad (1)
\]

\[
\sigma_t^2(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}) - \mathbf{k}_t(\mathbf{x})^T(\mathbf{K}_t + \sigma^2\mathbf{I})^{-1}\mathbf{k}_t(\mathbf{x}), \quad (2)
\]
In this work we also make use of a portfolio, or collection of base strategies, for Bayesian optimization. Although the Matérn tends to make less stringent smoothness assumptions, thus making it a good fit for Bayesian optimization.

In this work we also make use of a portfolio, or collection of base strategies, \( A = \{ \alpha_k \}_{k=1}^K \). Each strategy can be thought of as an expert which recommends that its candidate point \( x_{kt} = \alpha_k(t) \) be selected at iteration \( t \). We denote this point as \( x_k \) when the time index is unambiguous. Our task is to select the most promising candidate according to some meta-criterion. In particular, the approach we introduce directly utilizes a probabilistic model of the location of the unknown global minimizer \( x^* \). Given data \( D \), let

\[
\mathbb{P}(\mathrm{d}x_*|D) = \mathbb{P}\left( \arg \min x \in \mathcal{X} \in \mathrm{d}x_*|D \right) \tag{5}
\]

denote the posterior over minimizer locations, with density \( p(x_*|D) \). In Section 2.1 we propose our meta-criterion \( u(x_k|D) \) which corresponds to the expected entropy of the distribution (5) after selecting input \( x_k \). In other words the candidate selected by this criterion is the one that results in the greatest decrease in uncertainty (in terms of entropy) about the minimizer. We will also see that a crucial substep of this algorithm relies on sampling from (5). In Section 2.2 we show how to approximately sample from this distribution, an approach that also acts as an extension of Thompson sampling to continuous domains with GP priors. Finally, in Section 2.3 we summarize our selection strategy, the pseudocode for which is given in Algorithm 2. We also discuss how to effectively marginalize over GP hyperparameters within this strategy.

### 2.1 Entropy as a meta-criterion

We now consider a selection criterion that corresponds to the expected entropy that arises from selecting the recommendation of the \( k \)th acquisition strategy. This corresponds to the expectation

\[
\mathbb{E}_{p(y_k|D,x_k)}[H[p(x_*|\tilde{D}_k)]]
\]

where \( \tilde{D} = D \cup \{(x_k, y_k)\} \) is a random variable containing additional data distributed according to the posterior predictive distribution. For continuous densities \( p \) the differential entropy can be written as

\[
H[p(x)] = - \int p(x) \log p(x) \mathrm{d}x
\]

However, while the expectation with respect to the predictive distribution is easy to approximate via samples, the entropy computation itself is more complicated due to the difficulty of evaluating \( p(x_*|\tilde{D}_k) \).

Instead we will approximate this density with a discrete distribution \( \hat{p} \) restricted to a finite set of \textit{reprenter points} denoted \( \{z^{(i)}\}_{i=1}^G \) simulated from some alternative measure. Not only must the distribution of \( \{z^{(i)}\} \) share the same support as the \( p(x_*|\tilde{D}_k) \), but in order to obtain good performance it is desirable for it to closely match the target distribution. However, it turns out we have access to just such a distribution: in this work we will produce samples \( z^{(i)} \sim p(x_*|D) \). While this distribution is difficult to evaluate we can actually produce samples from it relatively easily—we provide details on how to do so in the next section.

We can now write our selection criterion as

\[
u(x_k|D) = \mathbb{E}_{p(y_k|D,x_k)}\left[ H[\hat{p}(x_*|\tilde{D}_k)] \right] \approx \frac{1}{N} \sum_{n=1}^N H[\hat{p}(x_*|\tilde{D}_k^{(n)})]
\]

where \( H[\cdot] \) now represents the discrete entropy of its argument. Note that the Monte Carlo approximation given above is the result of the \( N \) data terms \( \tilde{D}_k^{(n)} \) each of which includes a single additional sample \( y_k^{(n)} \sim p(\cdot|x_k,D) \). As a result of our discrete approximation we can now write

\[
H[\hat{p}(x_*|\tilde{D}_k^{(n)})] = - \sum_i \hat{p}(x_* = z^{(i)}|\tilde{D}_k^{(n)}) \log \hat{p}(x_*) = z^{(i)}|\tilde{D}_k^{(n)}) \tag{7}
\]

\( ^1 \)Technically the Matérn is a class of kernels; here we use a smoothness parameter of \( \frac{5}{2} \).
for which we are left with the problem of computing \( \hat{p}(x_\ast = z^{(i)}|\tilde{D}^{(n)}_k) \). Recall, however, that \( \hat{p} \) is the probability distribution over minimizers of a GP-distributed function where the minimizers are restricted to a finite set. This can easily be sampled exactly. Let the random variable \([f_{kn}] = f(z^{(i)})\), \(i = 1, \ldots, G\), be a vector of latent function values evaluated at the representer points and conditioned on data \( \tilde{D}^{(n)}_k \). This vector simply has a Gaussian distribution and as a result we can produce \( S \) samples \( f^{(i)}_{kn} \sim p(|\tilde{D}^{(n)}_k) \) from the resulting GP posterior. The probabilities necessary to compute the entropy can then be approximated by the relative counts

\[
\hat{p}_{ikn} = \frac{1}{S} \sum_s \mathbb{I}[i = \text{arg\,min}_j |f^{(s)}_{kn}|]
\]

that \( z^{(i)} \) is the minimizer among the sampled functions. Finally, by combining these ideas we can write our entropy-based meta-criterion as

\[
u(x_k|D) = -\frac{1}{N} \sum_{n=1}^N \sum_{i=1}^G \hat{p}_{ikn} \log \hat{p}_{ikn}.
\]

Pseudocode computing this quantity is given in Algorithm 3.

### 2.2 Sampling posterior global minimizers

The meta-criterion introduced in the previous section relied upon producing samples from (5), i.e. the posterior over global minima. While sampling from (5) is difficult in general, we can gain intuition by considering the finite domain setting. If the domain \( \mathcal{X} \) is restricted to a finite set of \( m \) points, the latent function \( f \) takes the form of an \( m \)-dimensional vector \( f \). The probability that the \( i \)th element of \( f \) is optimal can then be written as \( \int p(f|D) \prod_{j \neq i} \mathbb{I}[f_i \leq f_j] df \). This suggests the following generative process: i) draw a sample from the posterior distribution \( p(f|D) \) and ii) return the index of the maximum element in the sampled vector. This process is known as Thompson sampling or probability matching when used as an arm-selection strategy in multi-armed bandits [7]. This same approach could be used for sampling the maximizer over a continuous domain \( \mathcal{X} \). At first glance this would require constructing an infinite-dimensional object representing the function \( f \). To avoid this, one could sequentially construct \( f \) while it is being optimized. However, evaluating such an \( f \) would ultimately have cost \( O(m^3) \) where \( m \) is the number of function evaluations necessary to find the optimum. Instead, we propose to sample and optimize an analytic approximation to \( f \). We will briefly derive this approximation below.

Given a shift-invariant kernel \( k \), Bochner’s theorem [8] asserts the existence of its Fourier dual \( s(w) \), which is equal to the spectral density of \( k \). Letting \( p(w) = s(w)/\alpha \) be the associated normalized density, we can write the kernel as the expectation

\[
k(x, x') = \alpha E_p[w]e^{iwx^T(x - x')} = 2\alpha E_{p(w,b)}[\cos(w^T x + b) \cos(w^T x' + b)],
\]

where \( b \sim U[0, 2\pi] \). Let \( \phi(x) = \sqrt{2\alpha/m} \cos(Wx + b) \) denote an \( m \)-dimensional feature mapping and where \( W \) and \( b \) consist of \( m \) stacked samples from \( p(w, b) \). The kernel \( k \) can then be approximated by the inner product of these features, \( k(x, x') = \phi(x)^T \phi(x') \). This approach was used by [25] as an approximation method in the context of kernel methods. The feature mapping \( \phi(x) \) allows us to approximate the Gaussian process prior for \( f \) with a linear model \( f(x) = \phi(x)^T \theta \) where \( \theta \sim N(0, I) \) is a standard Gaussian. By conditioning on \( D \), the posterior for \( \theta \) is also multivariate Gaussian, \( \theta|D \sim N(A^{-1} \Phi^T y, \sigma^2 A^{-1}) \) where \( A = \Phi^T \Phi + \sigma^2 I \), \( y \) is the vector of the output data, and \( \Phi \) is a matrix of features evaluated on the input data.

Let \( \phi^{(i)} \) and \( \theta^{(i)} \) be a random set of features and the corresponding posterior weights sampled both according to the generative process given above. They can then be used to construct the function \( f^{(i)}(x) = \phi^{(i)}(x)^T \theta^{(i)} \), which is an approximate posterior sample of \( f \)—albeit one with a finite parameterization. We can then maximize this function to obtain \( z^{(i)} = \text{arg\,min}_x f^{(i)}(x) \), which is approximately distributed according to \( p(x|D) \). To produce samples from this process it is also necessary to derive the spectral density for any kernel of interest. This quantity is given by the kernel’s Fourier transform \( s(w) = \frac{1}{(2\pi)^m} \int e^{iw^T \tau} k(\tau, 0) d\tau \). For the squared-exponential and Matérn kernels we consider here this gives rise to weights which take the form

\[
w_{SE} \sim N(0, \text{diag}(\ell^2)^{-1}) \quad \text{and} \quad w_{MATÉRN} \sim T(0, \text{diag}(\ell^2)^{-1}, \frac{5}{2}),
\]

\( \ell \)
Consider now drawing \( M \) samples \( \{ \psi^{(i)} \} \) from the posterior \( p(\psi|D) \). Often an acquisition strategy is specified with respect to some internal acquisition function which is optimized at every iteration in order to select \( x_t \). These functions can then be approximately marginalized with respect to the hyperparameter samples in order to instead optimize an integrated acquisition function. This approach was taken in [27] for the EI and PI acquisition functions. Note that the randomized Thompson sam-

i.e. these are distributed as a normal and Student’s T respectively, which are easy to sample from.

Finally, while this approach can be used to produce samples which implement the entropy meta-criterion of the previous section, it can also be used directly as an acquisition strategy. Consider the randomized strategy given by

\[
\alpha_{\text{THOMPSON}}(D) = x^{(0)} \text{ where } x^{(0)} \sim p(x_1|D_1).
\]

Given the above description we can see that for finite \( \mathcal{X} \) this is exactly the Thompson sampling algorithm. However, our derivation extends this approach to continuously varying functions.

### 2.3 Algorithm summary

In Algorithm 1 we describe general pseudocode for performing Bayesian optimization using a portfolio of acquisition functions. Here \( \text{META} \text{POLICY} \) denotes a general selection criterion such as the GP-HEDGE approach of [13]. In Algorithm 2 we summarize ESP, the criterion developed in this work. The cost of this approach will be dominated by solving the linear system on line 6 of Algorithm 2. As a result the complexity will be on the order \( \mathcal{O}(KN^3t^3) \), however this is only a constant-factor slowdown when compared to standard Bayesian optimization algorithms and can easily be parallelized.

One consideration we have not mentioned is the selection of hyperparameters, which can have a huge effect on the performance of any Bayesian optimization algorithm. For example, if the length scale parameters \( \ell \) are chosen to be very small, then querying a point \( x \) will only allow the posterior to learn the model structure in a very small region around this point. If this parameter is chosen to be too small, this can greatly affect the optimization process as it will force the optimizer to uniformly explore small balls of width proportional to \( \ell \). For too large values of \( \ell \), the model will be too smooth and similar difficulties arise.

Typical approaches to GP regression will often optimize the marginal likelihood or “evidence” as a means of setting these parameters. However, in the Bayesian optimization setting this approach is particularly ineffective due to the initial paucity of data. Even worse, such optimization can also lead to quite severe local maxima around the initial data points. In this work we instead give a fully Bayesian treatment of the hyperparameters. Let \( \psi \) denote a vector of hyperparameters which includes any kernel and likelihood parameters. Let \( p(\psi|D) \propto p(\psi) p(D|\psi) \) denote the posterior distribution over these parameters where \( p(\psi) \) is a hyperprior and \( p(D|\psi) \) is the GP marginal likelihood. For a fully Bayesian treatment of \( \psi \) we must marginalize our acquisition strategy with respect to this posterior. The corresponding integral has no analytic expression and must be approximated using Monte Carlo.

Consider now drawing \( M \) samples \( \{ \psi^{(i)} \} \) from the posterior \( p(\psi|D) \). Often an acquisition strategy is specified with respect to some internal acquisition function which is optimized at every iteration in order to select \( x_t \). These functions can then be approximately marginalized with respect to the hyperparameter samples in order to instead optimize an integrated acquisition function. This approach was taken in [27] for the EI and PI acquisition functions. Note that the randomized Thompson sam-

---

**Algorithm 1 Portfolio Bayesian Optimization**

**Require:** a black-box with unknown mean \( f \), initial data \( D_0 = \{ \} \), a portfolio \( \mathcal{A} = \{ \alpha_k \} \)

1: \( t = 1, \ldots, T \) do
2: \( \text{collect candidates } x_{kt} = \alpha_k(D_{t-1}) \)
3: \( \text{set } x_t = \text{META} \text{POLICY}(\{ x_{kt} \}, D_{t-1}) \)
4: \( y_t = \text{BLACK} \text{BOX}(x_t) \)
5: \( \text{augment data } D_t = D_{t-1} \cup \{(x_t, y_t)\} \)
6: end for
7: return \( \bar{x}_T = \arg \max_{x \in \mathcal{X}} \mu_T(x) \)

**Algorithm 2 Entropy Search Portfolio**

**Require:** candidates \( \{ x_k \} \), observations \( D \)

1: \( z^{(i)} \sim p(x_i|D), \ i = 1, \ldots, G \)
2: for \( k = 1 : K \) do
3: for \( n = 1 : N \) do
4: \( y_{kn}^{(n)} \sim p(y|x_k, D) \)
5: \( \tilde{D}^{(n)} = D_t \cup \{(x_k, y_{kn}^{(n)})\} \)
6: \( \hat{f}^{(s)}_{kn} \sim p(f|\tilde{D}^{(n)}_k) \text{ for } s = 1 : S \)
7: \( \hat{p}_{ikn} = \frac{1}{S} \sum_s 1[i = \arg \min_j |f_{kn}^{(s)}_j|] \)
8: end for
9: \( u_k = \frac{1}{N} \sum_{n=1}^{N} \sum_{i=1}^{G} \hat{p}_{ikn} \log \hat{p}_{ikn} \)
10: end for
11: return \( x_k \), where \( k_\star = \arg \max_k u_k \)
pling strategy, introduced in Section 2.2, only requires a single hyperparameter sample as we can see this as a joint sample from the hyperparameters and the function minimizer.

Our approach has additional complexity in that our candidate selection criterion depends on the hyperparameter samples as well. This occurs explicitly in lines (4–6) of Algorithm 2, which sample from the GP posterior. This can be solved simply by adding an additional loop over the hyperparameter samples. Our particular use of replicator points in line 1 also depends on these hyperparameters. We can solve this problem by equally distributing our representers between the $M$ different hyperparameter samples.

## 3 Experiments

In this section we evaluate the proposed method ESP on several problems: two synthetic test functions commonly used for benchmarking in global optimization, two real-world automatic machine learning tasks, and a relatively high-dimensional simulated control problem. We compare our method against three well-known Bayesian optimization acquisition functions, namely the EI, PI, and THOMPSON methods described earlier. For EI we used the implementation available in the spearmint package, while the latter two were implemented in the same framework. Each of these methods was included in our portfolio for ESP. Note, we do not compare against GP-UCB as the bounds do not apply directly when integrating over GP hyperparameters. This would require an additional derivation which we do not consider here.

We also compare against the GP-HEDGE portfolio method of and an approach which randomly selects between different base strategies that we label RP. Finally, although not introduced in the text we also experimented with an alternative portfolio strategy which we label RÉNYI. The method was motivated by similar information-theoretic ideas as ESP and is based on a different measure of uncertainty, namely the Rényi $\alpha$-divergence. However, we found it to not perform well in practice so we include it here only for completeness.

In order to marginalize over the hyperparameters we used slice sampling as implemented in the spearmint package. By default, the spearmint software outputs 10 samples at each iteration. Both improvement-based methods, EI and PI, can simply be evaluated with each sample and averaged, while in keeping with the THOMPSON strategy, our THOMPSON implementation only uses a single sample, namely the last one. As reported above our proposed ESP method splits its 500 replicator points equally among the hyperparameter samples, i.e., for each of the 10 hyperparameters we draw 50 points. In addition, in order to estimate $u(x_k|D)$, for each of 5 simulated $y_{k}^{(n)}$ and each GP hyperparameter, ESP draws 1000 samples from $p(x_\star|\tilde{D}_{k}^{(n)})$, and then computes and averages the entropy estimates. Similar numbers of samples were used for the RÉNYI method.

In the following experiments we evaluate the performance of each algorithm as a function of the number of function evaluations. When the true optimum is known, as in the case of the Branin-Hoo and Hartmann 3 experiment, we measure performance by the absolute error and for all other experiments performance is given by the minimum function value obtained up to that iteration. Each experiment was repeated 20 times and for each method, we plot the median performance metric over these repetitions. Ultimately we are interested in a measure of the expected performance of each method. For readability, we omit the confidence bands, however these are shown in an additional plot for the repeller problem and are representative of the variance in all experiments.

### 3.1 Global optimization test functions

We begin with two synthetic functions commonly used for benchmarking global optimization methods: Branin-Hoo and Hartmann 3. They are two- and three-dimensional, respectively, and are both continuous, bounded, and multimodal. All 7 methods were run up to a final horizon of $T = 100$ and Figure 1 reports the observed performance measured in absolute error on a logarithmic scale.

Note that PI is known to be a greedy acquisition function but it can be tuned via a minimum improvement parameter. We left this parameter to its default value of 0 hence PI performs particularly poorly on these multimodal examples where being greedy can lead to suboptimal local minima. In
Figure 1: Absolute error of the best observation for the Branin and Hartmann 3 test functions. The four portfolio methods perform as well or better than the acquisition functions they include. At the final horizon on Branin, there is a full order of magnitude separating the best acquisition function and the worst portfolio. Interestingly, the poor performance of PI does not hinder the performance of ESP which converges fastest.

this work however we are interested in selecting between acquisition functions rather than tuning them. Therefore it is interesting to note that despite the poor performance of PI, all portfolios are still competitive with the highly optimized EI with ESP converging the fastest to errors of $10^{-6}$ and $10^{-8}$ on Branin and Hartmann 3, respectively. On the other hand, Rényi is being outperformed by ESP. Finally, notice that the naive random portfolio method RP performs well on average, and in fact out-performs the current state-of-the-art in acquisition portfolios.

### 3.2 Automatic machine learning: online LDA

Latent Dirichlet Allocation is a directed graphical model for documents used in topic modelling tasks, for which Hoffman et al. [12] proposed an online learning approach in the variational Bayes paradigm. In this experiment we use precomputed performance data for this online LDA algorithm on a dataset of 250,000 Wikipedia articles under many parameter settings. This performance dataset is publicly available [9]. Tuning the online LDA algorithm involves choosing the 2 learning parameters, $\tau_0$ and $\kappa$ as well as a third parameter specifying the mini-batch size, yielding a three-dimensional problem. Following the original authors, the search space is restricted to a $6 \times 6 \times 8$ grid [12].

Due to the discrete nature of this search space, quantifying uncertainty is slightly different from that presented in 2.1 so we also implemented a discrete version of ESP. This is the only information-based portfolio we consider for this and the next experiment. We ran the 6 methods up to a horizon of $T = 50$ function evaluations and our results are summarized in Figure 2.

In this example, we see that the median THOMPSON run did not converge to the optimum, and EI takes almost twice the number of evaluations to converge compared to RP. With the exception of PI, the portfolios once again out-perform the individual methods that comprise them. In this problem, it seems that being greedy actually benefits PI and by extension RP. Indeed, since RP selects candidates at random, PI is selected often. In contrast, being greedy does not provide information about the location of the optimum and so ESP does more exploration early on, which explains the slow start observed ESP’s performance. However we note that the method quickly catches up to its competitors and converges before GP-HEDGE and PI.

### 3.3 Automatic machine learning: structured SVM

In this experiment, we optimize the latent structured support vector machine (SVM) using a dataset available from [27,9]. Similar to the previous experiment, the authors consider the latent structured SVM on a three-dimensional grid of parameter settings which include two regularization parameters and a convergence tolerance. For this work we use data corresponding to the precomputed performance of this algorithm on a popular example task: the binary classification of protein DNA sequences [35,22].
The three-dimensional search space is discretized yielding a $25 \times 14 \times 4$ grid, once again following the original methodology. The result of the experiment is summarized in Figure 2. Similarly to the online LDA experiment, PI performs well—and in fact exceptionally so, compared to the synthetic examples. The RP and GP-HEDGE methods seem to benefit from exploiting PI’s good performance. Once again, ESP opts to explore the available space before exploiting but that exploration appears to pay off as ESP is able to catch up to the competing portfolio methods with a sharp drop at around 35 evaluations.

### 3.4 A control task

Our final experiment compares all 7 methods on a nine-dimensional control problem considered in previous work on Bayesian optimisation portfolios [13, 14]. A particle in a two-dimensional world is dropped from a fixed starting region and is accelerated downward by gravity. The task is to direct the falling particle through circular regions of low loss by placing 3 repellers with 3 degrees of freedom each, namely their position in the plane and repulsion strength. Figure 3 reports our results.

In this task, GP-HEDGE did not perform well, while RÉNYI fails to match the performance of EI. On the other hand, RP and ESP out-perform all other methods by a considerable margin under 40 evaluations and ESP continues to match the best portfolio acquisition functions thereafter.

### 4 Conclusion

In this work, we revisited the idea of portfolios for Bayesian optimization. We demonstrated empirically that our proposed information-theoretic ESP meta-criterion can indeed have robust behavior across functions of different dimensionality and modality even when the members of its portfolio do...
not exhibit this behavior. We also provided a mechanism for sampling representer points as a way of approximating $p(x_\star|D)$ which is more principled than previous approaches that were based on slice sampling from surrogate measures such as the expected improvement (EI). We additionally showed that this sampling mechanism can itself be used as an acquisition strategy, thereby extending the popular Thompson sampling approach to continuous spaces with kernel methods. This technique had previously been confined to finite domains. Finally, we show that even the naive approach of random selection among acquisition functions within a portfolio can yield both performance gains and robustness.

References

[1] J. Azimi, A. Jalali, and X. Fern. Hybrid batch bayesian optimization. In ICML, 2012.
[2] 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.
[3] S. Bochner. Lectures on Fourier integrals. Princeton University Press, 1959.
[4] E. Brochu, T. Brochu, and N. de Freitas. A Bayesian interactive optimization approach to procedural animation design. In Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pages 103–112, 2010.
[5] E. Brochu, V. M. Cora, and N. de Freitas. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. Technical Report UBC TR-2009-23 and arXiv:1012.2599v1, Dept. of Computer Science, University of British Columbia, 2009.
[6] E. Brochu, N. de Freitas, and A. Ghosh. Active preference learning with discrete choice data. In Advances in Neural Information Processing Systems, pages 409–416, 2007.
[7] O. Chapelle and L. Li. An empirical evaluation of Thompson sampling. In Advances in Neural Information Processing Systems, pages 2249–2257, 2011.
[8] N. de Freitas, A. Smola, and M. Zoghi. Exponential regret bounds for Gaussian process bandits with deterministic observations. In International Conference on Machine Learning, 2012.
[9] K. Eggensperger, M. Feurer, F. Hutter, J. Bergstra, J. Snoek, H. Hoos, and K. Leyton-Brown. Towards an empirical foundation for assessing Bayesian optimization of hyperparameters. In NIPS Workshop on Bayesian Optimization in Theory and Practice, 2013.
[10] R. Garnett, M. A. Osborne, and S. J. Roberts. Bayesian optimization for sensor set selection. In ACM/IEEE International Conference on Information Processing in Sensor Networks, pages 209–219. ACM, 2010.
[11] P. Hennig and C. Schuler. Entropy search for information-efficient global optimization. The Journal of Machine Learning Research, 98888:1809–1837, 2012.
[12] M. D. Hoffman, F. R. Bach, and D. M. Blei. Online learning for latent Dirichlet allocation. In advances in neural information processing systems, pages 856–864, 2010.
[13] M. W. Hoffman, E. Brochu, and N. de Freitas. Portfolio allocation for Bayesian optimization. In Uncertainty in Artificial Intelligence, pages 327–336, 2011.
[14] M. W. Hoffman, H. Kueck, N. de Freitas, and A. Doucet. New inference strategies for solving Markov decision processes using reversible jump MCMC. In Uncertainty in Artificial Intelligence, pages 223–231, 2009.
[15] 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.
[16] F. Hutter, H. H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In LION, pages 507–523, 2011.
[17] D. Lizotte. Practical Bayesian Optimization. PhD thesis, University of Alberta, Canada, 2008.
[18] D. Lizotte, T. Wang, M. Bowling, and D. Schuurmans. Automatic gait optimization with Gaussian process regression. In Proc. of IJCAI, pages 944–949, 2007.
[19] 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.
[20] R. Marchant and F. Ramos. Bayesian optimisation for intelligent environmental monitoring. In NIPS workshop on Bayesian Optimization and Decision Making, 2012.
[21] R. Martinez-Cantin, N. de Freitas, A. Doucet, and J. A. Castellanos. Active policy learning for robot planning and exploration under uncertainty. Robotics Science and Systems, 2007.
[22] K. Miller, M. P. Kumar, B. Packer, D. Goodman, D. Koller, et al. Max-margin min-entropy models. In AISTATS, 2012.

[23] 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.

[24] D. Pal, B. Poczos, and C. Szepesvari. Estimation of Renyi entropy and mutual information based on generalized nearest-neighbor graphs. In Advances in Neural Information Processing Systems, pages 1849–1857, 2010.

[25] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In Advances in Neural Information Processing Systems, pages 1177–1184, 2007.

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

[27] 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.

[28] 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.

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

[30] C. Thornton, F. Hutter, H. H. Hoos, and K. Leyton-Brown. Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms. In Knowledge Discovery and Data Mining, pages 847–855, 2013.

[31] 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.

[32] Z. Wang, S. Mohamed, and N. de Freitas. Adaptive Hamiltonian and Riemann manifold Monte Carlo samplers. In International Conference on Machine Learning, pages 1462–1470, 2013.

[33] Z. Wang, B. Shakibi, L. Jin, and N. de Freitas. Bayesian multi-scale optimistic optimization. In AI and Statistics, pages 1005–1014, 2014.

[34] Z. Wang, M. Zoghi, D. Matheson, F. Hutter, and N. de Freitas. Bayesian optimization in high dimensions via random embeddings. In International Joint Conference on Artificial Intelligence, pages 1778–1784, 2013.

[35] C.-N. J. Yu and T. Joachims. Learning structural SVMs with latent variables. In Proceedings of the 26th Annual International Conference on Machine Learning, pages 1169–1176. ACM, 2009.