Entropy Search for Bayesian Optimisation Based on Parabolic Approximation

University Of Oxford

MEng Engineering Science

Author: Binxin Ru, St Edmund Hall
Supervisor: Prof. Michael Osborne

Trinity Term 2016

Acknowledgements

I would like to express my greatest gratitude to my supervisor, Prof. Michael Osborne, for the inspiration, guidance and encouragement. It was really an enriching journey for me to work under your supervision and thank you for motivating and enlightening me when I was confused or depressed by the experimental results. I would also like to thank several Ph.D candidates in the Machine Learning Research Group. First, I'm really grateful of Jack Fitzsimons who has provided me with highly informative and systematic supervision during my summer research project, enabling me to develop a much more solid foundation in the field of machine learning. Second, I would like to thank Mark McLeod and Favour Mandanji Nyikosa for helping me better understand Predictive Entropy Search and helping me tackle some technical issues in implementing the algorithms. Finally, I want to thank Gabriele Abbati for discussing entropy based methods with me and helping me install some C packages.

Abstract

Bayesian optimisation is a powerful technique for optimising black-box functions that are expensive to evaluate[1]. A popular category of Bayesian optimisation techniques is based on information theory and recommends evaluation locations by maximising the information gain about the global minimiser of the objective function. Predictive Entropy Search (PES) is an important representative of information-based techniques and has demonstrated good empirical performance. However, PES requires many approximations to implement and face several drawbacks in its applications. We develop a novel information-based alternative, namely Entropy Search for Bayesian optimisation based on parabolic approximation (ESBOPA), that smartly approximates the unknown objective function in a parabolic form and represent the global minimum explicitly with a hyperparameter η. In comparison with PES, our technique faces less constraints on the choice of appropriate covariance functions for the Gaussian process prior, works in output space instead of input space as well as circumvents the need for sampling global minimiser in entropy computation. We demonstrate that ESBOPA performs as well as PES and other popular Bayesian optimisation techniques in several empirical tests.

Contents

1 Introduction

Optimisation problems arise in numerous fields ranging from science and engineering to economics and management. Such problem can be expressed in a mathematical form:

$$\min_{\mathbf{x}\in\chi} f(\mathbf{x}) \tag{1.1}$$

where χ is the constrained domain of interest and $f(\mathbf{x})$ is the objective function. Usually we deal with classical optimisation problems in which the objective function is known together with its derivatives or the function is cheap to evaluate [12]. However, in many situations, we face another type of optimisation problems for which the above assumptions do not apply. For example, in the cases of clinical trials, financial investments or constructing a sensor network, it is very costly to draw a sample from the latent function underlying the real-world processes. Thus, efficient use of data available is very important. The objective functions in such type of problems are also non-convex in general and act like black boxes whose closed-form expressions and derivatives are unknown [1]. Bayesian optimisation is a powerful tool to tackle such optimisation challenges.

A core step in Bayesian optimisation is to definite an acquisition function which uses the available observations effectively to recommend the next query location. There are many types of acquisition functions ranging from improvement-based measures, such as Probability of Improvement [6] and Expected Improvement [**?**], to optimistic measures, such as Gaussian Process Upper Confidence Bound [11]. The most recent type, which is based on information theory, offers a brand new perspective to efficiently select the sequence of sampling locations based on entropy of the distribution over the unknown minimiser $\mathbf{x}_*$ [2]. The information-based approaches guide our evaluations to locations where we can maximise our learning about the unknown minimum rather than to locations where we expect to obtain lower functional values[12]. However, the existing information-based acquisition functions, mainly Entropy Search and Predictive Entropy Search, require many approximations to implement and face serious constraints in its application. This project aims to develop a novel information-based approach which has extended applicability and is much easier to implement. Specifically, our approach has three major advantages over the state-of-the-art technique (i.e. Predictive Entropy Search). First, our approach faces less constraints on the choice of appropriate covariance functions for the Gaussian process prior. Second, our approach focuses on the information content in output space instead of input space, thus more efficient in high dimensional problems. Third, our approach circumvents the need to generate minimiser samples for entropy computation, thus saving much efforts for the sampling process. Because of its reliance on the smart use of the parabolic expression, our novel acquisition function is called entropy search for Bayesian optimisation based on parabolic approximation (ES-

BOPA).

This report is organised into three parts. To begin with, Chapter 2 provides a literature review on the relevant knowledge about Bayesian optimisation. The review covers the choice of the prior distribution for Bayesian optimisation and a general overview of several popular types of acquisition functions. Particularly, Predictive Entropy Search is discussed in detail because it is the state-of-the-art entropy-based approach. Chapter 3 focuses on the technical details of our ESBOPA approach. It starts with the core idea of approximating any objective function with a parabolic form to directly represent the global minimum as a hyperparameter η and the use of local linearisation to make the approximation analysis tractable. The second section of the chapter goes through the derivation of ESBOPA approach as well as the methods adopted to compute various components of ESBOPA acquisition function for implementation. This is followed by a discussion on the possible ways to treat hyperparameters in our inference. Finally, the series of experiments we used to test ESBOPA approach are explained in Chapter 4 and the performance of ESBOPA is compared with those of EI and PI.

2 Literature Review

2.1 Bayesian Optimisation

Bayesian optimisation is a power tool for finding the optimum of a black-box objective function which is costly to evaluate. It is particularly useful in the cases when the function does not have an analytical form but noisy observations of the function at sampled points can be obtained [1].

To perform Bayesian optimisation, we first decide a prior belief over the unknown objective functions and then update this prior model by incorporating observation data to obtain a Bayesian posterior. The posterior reflects our updated beliefs about the objective function based on data observed, . In order to take the next sample efficiently, we compute an acquisition function which harnesses the mean and variance of the posterior distribution to evaluate the utility of potential locations for the next evaluation. The next query location is then determined by maximising the acquisition function [2]. The procedures of Bayesian optimisation is summarised in Algorithm 1 [1]. And Figure **??** demonstrates three iterations of Bayesian optimisation on a 1D objective function.

Algorithm 1 Bayesian optimisation

```
for n=1,... do
    select new $\mathbf{x}_{n+1}$ by optimising acquisition function $\alpha : \mathbf{x}_{n+1} = \arg\max_{\mathbf{x}} \alpha(\mathbf{x}|D_n)$
    evaluate the objective function to obtain $y_{n+1}$
    augment observation data $D_{n+1} = \{D_n, (\mathbf{x}_{n+1}, y_{n+1})\}$
    update statistical model
end for
```

In essence, Bayesian optimisation approach has two key components: 1) the prior distribution that represents our belief on the unknown objective function and 2) the acquisition function that evaluates the utility of possible query points and recommend the next point that would lead to utility maximisation.

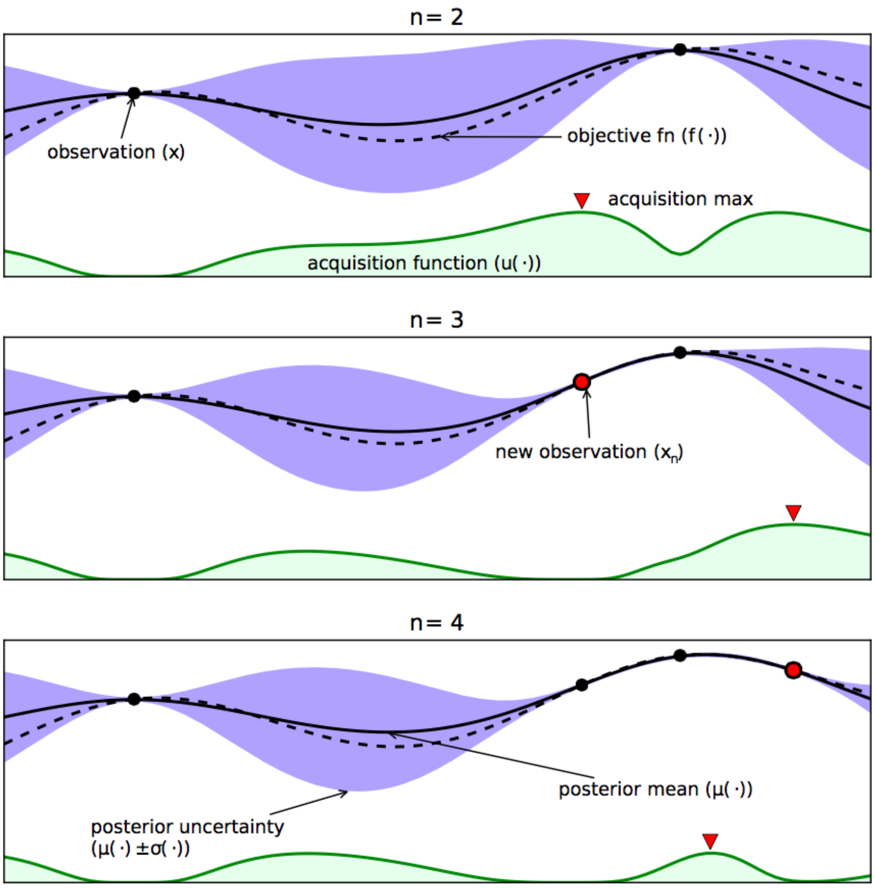

Figure 2.1: *Demonstration of the use of Bayesian optimisation on a 1D example. The figure shows the mean and confidence intervals of the Gaussian process model that approximates the objective function. The acquisition function (green) is high where the Gaussian process model predicts a high objective (exploitation) and where the prediction uncertainty is high (exploration). It is interesting to note that the region on the far left remains unsampled because while it has high uncertainty, it is correctly predicted to offer little improvement over the highest observation [1]*

2.1.1 The Gaussian Process Prior

Bayesian optimisation requires the definition of a prior distribution. When the number of observation is large, a prior distribution can enable convergence to the global optimum of the objective function if 1) the acquisition function is continuous and approximately minimises the expected deviation from the global minimum at a query point and 2) the conditional variance converges to zero at the observation data [3]. Many prior models fulfil these conditions but in this project, we choose to use the Gaussian process which is a popular non-parametric model.

The Gaussian process is a collection of random variables, any finite number of which have a multivariate Gaussian distribution [4]. It is fully specified by a mean function m and a covariance function k [4]:

$$f(\mathbf{x}) \sim \mathcal{GP}\big(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\big). \tag{2.1}$$

At an arbitrary input $\mathbf{x}$, a Gaussian process returns the mean and variance of a normal distribution over the possible values of $f(\mathbf{x})$ [1]. The mean function of the prior distribution provides an offset and specifies our inference far from observations [?]. For simplicity, the mean function is usually set to zero [4] and thus, the Gaussian process prior is solely determined by the covariance function.

By sampling the objective function at n input locations, we obtain observation data $D_f = \{(\mathbf{x}_i, f_i)|i = 1, \dots, n\} = \{(\mathbf{X}_n, \mathbf{f}_n)\}$. The function values $\mathbf{f}_n$ are drawn based on the multivariate normal distribution $p(\mathbf{f}_n) = \mathcal{N}\big(\mathbf{f}_n; \mathbf{0}, K(\mathbf{X}_n, \mathbf{X}_n)\big)$, where each element of the covariance matrix K is defined as $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$.

The joint distribution of the observed function values $\mathbf{f}_n$ and the function value at a test location $f = f(\mathbf{x})$ is also Gaussian [4] :

$$\begin{bmatrix} \mathbf{f}_n \\ f \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} K(\mathbf{X}_n, \mathbf{X}_n) & K(\mathbf{X}_n, \mathbf{x}) \\ K(\mathbf{x}, \mathbf{X}_n) & K(\mathbf{x}, \mathbf{x}) \end{bmatrix}\right). \tag{2.2}$$

By manipulating the joint distribution, we obtain the predictive posterior distribution at the test point $\mathbf{x}$:

$$p(f|\mathbf{x}, \mathbf{X}_n, \mathbf{f}_n) = \mathcal{N}\big(f; m_f(\cdot), K_f(\cdot, \cdot)\big) \tag{2.3}$$

where

$$m_f(\mathbf{x}) = K(\mathbf{x}, \mathbf{X}_n) K(\mathbf{X}_n, \mathbf{X}_n)^{-1} \mathbf{f}_n \tag{2.4}$$

$$K_f(\mathbf{x}, \mathbf{x}') = K(\mathbf{x}, \mathbf{x}') - K(\mathbf{x}, \mathbf{X}_n) K(\mathbf{X}_n, \mathbf{X}_n)^{-1} K(\mathbf{X}_n, \mathbf{x}). \tag{2.5}$$

However, in real word situation, we do not have access to the true function values but only noisy observation of the function $y(\mathbf{x}_i) = f(\mathbf{x}_i) + \epsilon$ where ϵ is assumed to be an independently and identically distributed Gaussian noise with variance σ_n^2 $\big($i.e. $p(\mathbf{y}|\mathbf{f}) = \mathcal{N}(\mathbf{f}, \sigma_n^2 \mathbf{I})\big)$ [4]. In the case of noisy observation data $D_n = \{(\mathbf{x_i}, y_i)|i = 1, \dots n\} = \{\mathbf{X}_n, \mathbf{y}_n\}$, the Gaussian process prior becomes $p(\mathbf{y}_n) = \mathcal{N}\big(\mathbf{y}_n; \mathbf{0}, K(\mathbf{X}_n, \mathbf{X}_n) + \sigma_n^2 \mathbf{I}\big)$. The predictive distribution of the noisy observation y at the test point $\mathbf{x}$ has the form

$$p(y|\mathbf{x}, \mathbf{X}_n, \mathbf{f}_n) = \mathcal{N}\big(y; m_y(\cdot), K_y(\cdot, \cdot)\big) \tag{2.6}$$

where

$$m_y(\mathbf{x}) = K(\mathbf{x}, \mathbf{X}_n) \big[K(\mathbf{X}_n, \mathbf{X}_n) + \sigma_n^2 \mathbf{I}\big]^{-1} \mathbf{y}_n \tag{2.7}$$

$$K_y(\mathbf{x}, \mathbf{x}') = K(\mathbf{x}, \mathbf{x}') - K(\mathbf{x}, \mathbf{X}_n) \big[K(\mathbf{X}_n, \mathbf{X}_n) + \sigma_n^2 \mathbf{I}\big]^{-1} K(\mathbf{X}_n, \mathbf{x}). \tag{2.8}$$

2.1.1.1 Covariance Functions

The choice of covariance function is essential for a Gaussian process because it affects the smoothness properties of the Gaussian process prior [1] and thus determines the structure of the response functions we can model [2]. A covariance function is valid if its Gram matrix K whose element is $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ is positive semidefinite and valid covariance functions can be expressed as an inner product in the feature space $k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$ [5]. New valid covariance functions can be efficiently constructed by manipulating proven simple covariance functions (e.g. multiplying and/or adding simple covariances) [4].

In this project, we use the squared exponential covariance function with automatic relevance determination (SE-ARD), which is a popular type of covariance functions that allows us to learn the characteristic length scales in each input dimension. Characteristic length scales measure the distance we need to move in input space for the function values to be uncorrelated [4]. If the characteristic length scale of a certain input is very large, the covariance will become almost independent of that input as shown in Figure **??**.

The analytical form of the squared exponential ARD covariance function is

$$k(\mathbf{x}_i, \mathbf{x}_j) = \sigma^2 \exp\left[-\frac{1}{2}(\mathbf{x}_i - \mathbf{x}_j)^T \Lambda_l^{-2} (\mathbf{x}_i - \mathbf{x}_j)\right] \tag{2.9}$$

where σ is the output scale and Λ_l is a diagonal matrix of D length scales.

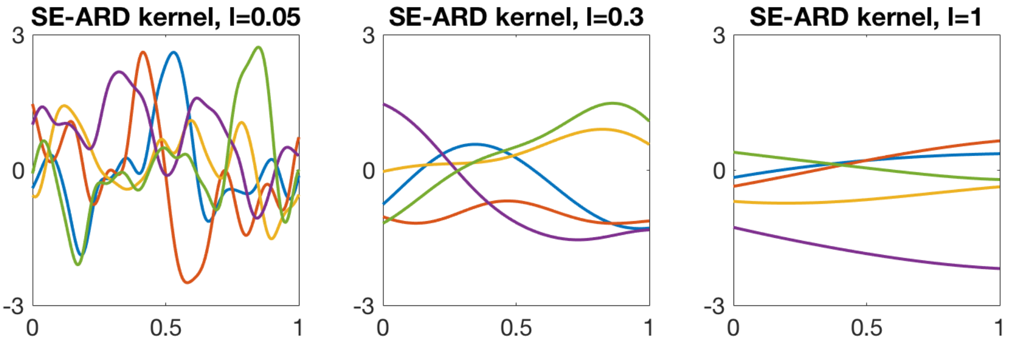

Figure 2.2: *Random samples generated by a Gaussian process prior with the SE-ARD covariance function and different characteristic lengths.*

2.1.2 Acquisition Functions

Acquisition functions are carefully designed to balance exploration and exploitation. Thus, they would recommend the next query location where the function value predicted by the posterior distribution is high (exploitation) and/or the uncertainty is large (exploration) [1]. As compared to the objective function, the acquisition functions must be much cheaper to evaluate or approximate and thus can be optimised more easily [2]. In this section, we would introduce three main categories of acquisition functions: 1) improvement-based acquisition functions, 2) optimistic acquisition functions and 3) the information-based acquisition functions.

2.1.2.1 Improvement-based Acquisition Functions

Probability of Improvement

An simple form of acquisition function developed early in 1964 is Probability of Improvement (PI) [6]. PI method measures the probability that a location $\mathbf{x}$ leads to an improvement upon the incumbent maximum value observed $f(\hat{\mathbf{x}})$ where $\hat{\mathbf{x}} = \arg\max_{\mathbf{x}_i} f(\mathbf{x}_i)$ for $i = 1, \ldots, n$. This probability is computed in the form of a cumulative distribution function[1]:

$$\alpha_{PI}(\mathbf{x}|D_n) = Prob(f(\mathbf{x}) \geq f(\hat{\mathbf{x}})) = \Phi\left(\frac{m(\mathbf{x}) - f(\hat{\mathbf{x}})}{\sigma(\mathbf{x})}\right) \tag{2.10}$$

where Φ is the standard normal cumulative distribution and $m(\mathbf{x})$ and $\sigma(\mathbf{x})$ are the mean and standard deviation of the posterior distribution at a test location $\mathbf{x}$. The next sample is then taken at the location with the maximum $\alpha_{PI}(\mathbf{x}|D_n)$.

PI acquisition function tends to be highly exploitation-oriented. To encourage more exploration, Kushner [6] proposes to add a positive trade-off parameter ζ so that PI method would recommend the next point by maximising the probability of improving over $f(\hat{\mathbf{x}}) + \zeta$. However, the value of ζ is often set arbitrarily. If ζ is too small, the search will be highly local whereas if ζ is too high, the search will be global but slow to converge [7].

Expected Improvement

Expected Improvement (EI) acquisition function offers an alternative to PI method by considering the magnitude of improvement[8] [?]. In EI acquisition function, the utility is represented by an improvement function $I(\mathbf{x})$ which is defined as

$$I(\mathbf{x}) = max\{0, f(\mathbf{x}) - f(\hat{\mathbf{x}})\}. \tag{2.11}$$

$I(\mathbf{x})$ is positive only when the prediction at the test location leads to an improvement over the maximum value known thus far $f(\hat{\mathbf{x}})$. Otherwise, $I(\mathbf{x})$ is zero. The new query point is then recommended by maximising the expected improvement

$$\mathbf{x} = \arg\max_{\mathbf{x}} \mathbb{E}[I(\mathbf{x})|D_n]. \tag{2.12}$$

The expectation can be computed analytically [8]:

$$\begin{aligned}
\alpha_{EI}(\mathbf{x}|D_n) &= \mathbb{E}[I(\mathbf{x})|D_n] \\
&= \int_0^\infty I(\mathbf{x})\mathcal{N}\big(I; m(\mathbf{x}) - f(\hat{\mathbf{x}}), \sigma(\mathbf{x})^2\big)\mathrm{d}I(\mathbf{x}) \\
&= \begin{cases} \big(m(\mathbf{x}) - f(\hat{\mathbf{x}})\big)\Phi\left(\dfrac{m(\mathbf{x}) - f(\hat{\mathbf{x}})}{\sigma(\mathbf{x})}\right) + \sigma(\mathbf{x})\phi\left(\dfrac{m(\mathbf{x}) - f(\hat{\mathbf{x}})}{\sigma(\mathbf{x})}\right) & \text{if} \quad \sigma(\mathbf{x}) > 0 \\ 0 & \text{if} \quad \sigma(\mathbf{x}) = 0 \end{cases}
\end{aligned}$$

where Φ and ϕ denotes the cdf and pdf of the standard normal distribution, and $m(\mathbf{x})$ and $\sigma(\mathbf{x})$ are the mean and standard deviation of the posterior distribution at a test location $\mathbf{x}$.

The two terms in EI acquisition function balances the trade-off between exploitation and exploration respectively. Similar to the PI case, Lizotte [9] proposes to incorporate a parameter ζ into EI acquisition function to control the trade-off between local optimisation and global search. Lizotte's experiments suggeste that $\zeta = 0.01$ works well in almost all cases.

2.1.2.2 Optimistic Acquisition Functions

Upper Confidence Bound

The concept of using confidence bounds for Bayesian optimisation is first explored by Cox and John [10] in their Sequential Design for Optimisation(SDO) algorithm. SDO recommends the next point for evaluation by maximising the following acquisition function:

$$\alpha_{UCB}(\mathbf{x}|D_n) = m(\mathbf{x}) + \kappa\sigma(\mathbf{x}). \tag{2.13}$$

which is a weighted sum of posterior mean $m(\mathbf{x})$ and standard deviation $\sigma(\mathbf{x})$, thus incorporating the trade-off between exploitation and exploration. κ is a parameter set by the user.

Srinivas et al. [11] then propose a guideline for setting and scheduling the value of κ which can lead to minimisation of the cumulative regret:

$$R_n = \sum_{i=1}^{n} f(\mathbf{x}_*) - f(\mathbf{x}_i) \tag{2.14}$$

where $\mathbf{x}_*$ is the true global maximiser and $\mathbf{x}_i$ is the query point selected via the acquisition function. With this guideline, they define the Gaussian Process Upper Confidence Bound (GP-UCB) whose acquisition function is a modified version of Equation 2.13:

$$\alpha_{GP-UCB}(\mathbf{x}|D_n) = m(\mathbf{x}) + \sqrt{\upsilon\tau_n}\sigma(\mathbf{x}) \tag{2.15}$$

where $\upsilon > 0$ and $\tau_n = 2\log(n^{d/2+2}\pi^2/3\delta)$. d is the input dimension and n is the number of observation taken.

2.1.2.3 Information-based Acquisition Function

Entropy Search

Entropy Search (ES) [12] aims to reduce the uncertainty about the unknown global minimiser $\mathbf{x}_*$ by selecting a query point that leads to the largest reduction in entropy of the distribution $p(\mathbf{x}_*|D_n)$. The acquisition function for ES has the form [12]:

$$\alpha_{ES}(\mathbf{x}|D_n) = H[p(\mathbf{x}_*|D_n)] - \mathbb{E}_{p(y|D_n,\mathbf{x})}\Big[H\big[p\big(\mathbf{x}_*|D_n \cup (\mathbf{x},y)\big)\big]\Big] \tag{2.16}$$

where $H[p(\mathbf{x})] = -\int p(\mathbf{x})\log p(\mathbf{x})\mathrm{d}\mathbf{x}$ and the expectation is taken with respect to the posterior distribution $p(y|D_n,\mathbf{x})$.

The first differential entropy term in the above function embodies our current uncertainty about the unknown minimiser $\mathbf{x}_*$. The second term measures our expected uncertainty about $\mathbf{x}_*$ after querying an arbitrary point $(\mathbf{x}, y)$. Thus, the acquisition function (Equation 2.16), which is the difference between the two entropy terms, represents the information gain about $\mathbf{x}_*$ and ES method would select the point that maximise this information gain.

However, an exact evaluation of the ES acquisition function (Equation 2.16) is only feasible after many approximations because the entropy terms cannot be computed analytically and the optimisation of Equation 2.16 involves calculating $p\big(\mathbf{x}_*|D_n \cup (\mathbf{x},y)\big)$ for many different values of $\mathbf{x}$ and y [13]. The approximation proposed by [12] is based on discretisation of the continuous search space, which results in a high computational costs of $O(M^4)$ where M is the number of discrete representer points.

Predictive Entropy Search

In view of the difficulties of implementing ES method, Lobato et al.[13] propose a modified alternative, named Predictive Entropy Search(PES). PES method utilises the symmetric property of mutual information and rewrite Equation 2.16 as [13] :

$$\alpha_{PES}(\mathbf{x}|D_n) = H[p(y|D_n,\mathbf{x})] - \mathbb{E}_{p(\mathbf{x}_*|D_n)}[H[p(y|D_n,\mathbf{x},\mathbf{x}_*)]] \tag{2.17}$$

where $p(y|D_n, \mathbf{x}, \mathbf{x}_*)$ is the posterior distribution for y conditioned on the observation data D_n, the test point $\mathbf{x}$ and the global maximiser $\mathbf{x}_*$. Different from ES acquisition function(Equation 2.16) which depends on the entropies of distributions on maximiser $\mathbf{x}_*$, the PES acquisition function (Equation 2.17) uses the entropies of predictive posterior distributions over output y which can be computed analytically or be approximated more easily [13].

PES has been shown to achieve better optimisation performance than ES while requiring less computational costs to be implemented. Thus, PES is the state of the art in information-based approaches. We will discuss PES method in more details in Section 2.1.3 because our project takes PES as a basis and develops a novel information-based approach, named Entropy Search for Bayesian optimisation based on quadratic approximation (ESBOPA).

2.1.3 Predictive Entropy Search

2.1.3.1 Algorithm

As mentioned above, the acquisition function for PES has the form:

$$\alpha_{PES}(\mathbf{x}|D_n) = H[p(y|D_n, \mathbf{x})] - \mathbb{E}_{p(\mathbf{x}_*|D_n)}[H[p(y|D_n, \mathbf{x}, \mathbf{x}_*)]]. \tag{2.18}$$

The first entropy term in Equation 2.18 has closed-form as the posterior distribution is a Gaussian:

$$H[p(y|D_n, \mathbf{x})] = 0.5 \log[2\pi e(v_n(\mathbf{x}) + \sigma_n^2)] \tag{2.19}$$

where $v_n(\mathbf{x})$ is the variance of the posterior distribution at a test point $\mathbf{x}$ which can be computed using Equation 2.5 and σ_n^2 is the variance of observation noise. The second term cannot be computed analytically and we need sophisticated techniques to approximate the expectation, the predictive posterior distribution $p(y|D_n, \mathbf{x}, \mathbf{x}_*)$ as well as the entropy of this distribution [13].

First, the expectation can be approximated by drawing Thompson samples of maximisers $\mathbf{x}_*$ from $p(\mathbf{x}_*|D_n)$ and then compute the average entropy over all samples [13]. According to Bochner's theorem [14], any continuous stationary covariance function k has a corresponding spectral density $s(\mathbf{w})$ obtained by Fourier transform. Using the normalised version of the spectral density $p(\mathbf{w}) = s(\mathbf{w})/\alpha$, the covariance function can be re-expressed as

$$k(\mathbf{x}, \mathbf{x}') = \alpha \mathbb{E}_{p(\mathbf{w})}\left[\exp[-i\mathbf{w}^T(\mathbf{x} - \mathbf{x}')]\right] = 2\alpha \mathbb{E}_{p(\mathbf{w},b)}\left[\cos(\mathbf{w}^T\mathbf{x} + b)\cos(\mathbf{w}^T\mathbf{x}' + b)\right] \tag{2.20}$$

where $b \sim \mathcal{U}[0, 2\pi]$ [15]. By using an m-dimensional feature vector $\phi(\mathbf{x}) = \sqrt{2\alpha/m}\cos(\mathbf{W}\mathbf{x} + \mathbf{b})$ where $\mathbf{W}$ and $\mathbf{b}$ are m stacked samples from $p(\mathbf{w}, b)$, we can rewrite the covariance function as $k(\mathbf{x}, \mathbf{x}') \approx \phi(\mathbf{x})^T\phi(\mathbf{x}')$ and approximate the prior for the objective function with a linear model $f(\mathbf{x}) =$

$\phi(\mathbf{x})^T\boldsymbol{\psi}$[13]. $\boldsymbol{\psi}$ is a standard multivariate Gaussian whose posterior distribution is also a Gaussian:

$$\boldsymbol{\psi} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \quad \text{and} \quad \boldsymbol{\psi}|D_n \sim \mathcal{N}\left((\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \sigma_n^2\mathbf{I})^{-1}\boldsymbol{\Phi}^T\mathbf{y}_n, \sigma_n^2(\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \sigma_n^2\mathbf{I})^{-1}\right)$$

where $\boldsymbol{\Phi}^T = [\boldsymbol{\phi}(\mathbf{x}_1)\dots\boldsymbol{\phi}(\mathbf{x}_n)]$ [13]. Now we can sample the maximiser $\mathbf{x}_*^{(i)}$ by minimising the approximated posterior sample of $f^{(i)}$:

$$\mathbf{x}_*^{(i)} = \arg\min_{\mathbf{x}} f^{(i)}(\mathbf{x}) = \arg\min_{\mathbf{x}} \phi^{(i)}(\mathbf{x})^T\boldsymbol{\psi}^{(i)}. \tag{2.21}$$

Another key component of PES acquisition function is the predictive posterior distribution which can be computed via marginalisation: $p(y|D_n, \mathbf{x}, \mathbf{x}_*) = \int p\big(y|f(\mathbf{x})\big)p\big(f(\mathbf{x})|D_n, \mathbf{x}, \mathbf{x}_*\big)\mathrm{d}f(\mathbf{x})$. The likelihood function is a Gaussian $p\big(y|f(\mathbf{x})\big) = \mathcal{N}\big(y; f(\mathbf{x}), \sigma_n^2\big)$. The posterior distribution $p\big(f(\mathbf{x})|D_n, \mathbf{x}, \mathbf{x}_*\big)$ is intractable because it is conditioned on the global maximiser $\mathbf{x}_*$. Lobato et al.[13] overcome this intractability by using three simplified constraints (C1,C2,C3) and approximate the posterior distribution in the form: $p(f(\mathbf{x})|D, \mathbf{x}, \mathbf{x}_*) = p(f(\mathbf{x})|D, C1, C2, C3)$.

The three constraints are as follows:

C1. $\mathbf{x}_*$ is a local maximiser. This is satisfied if $\bigtriangledown f(\mathbf{x}_*) = \mathbf{0}$ (C 1.1) and $\bigtriangledown^2 f(\mathbf{x}_*)$ is negative definite (C 1.2). Lobato et al.[13] further simplify (C 1.2) by assuming that the non-diagonal elements of $\bigtriangledown^2 f(\mathbf{x}_*)$, denoted as upper $[\bigtriangledown^2 f(\mathbf{x}_*)]]$, are zeros. Thus, the constraint C 1.2 becomes that the diagonal elements of the Hessian matrix $[\bigtriangledown^2 f(\mathbf{x}_*)]$ are negative.

C2. $f(\mathbf{x}_*)$ is larger than current observations. This implies $f(\mathbf{x}_*) \geq f(\mathbf{x}_i)$ for $i \leq n$. However, in reality, we only have noisy observations y_i. To avoid performing inference on the true function values, the above inequality is approximated with a looser constraint $f(\mathbf{x}_*) > y_{max} + \epsilon$ where ϵ is a zero-mean Gaussian noise with variance σ_n^2 and y_{max} is the largest observation.

C3. $f(\mathbf{x}_*)$ is greater than $f(\mathbf{x})$. This constraint means that we will consider the test point location $\mathbf{x}$ only if $f(\mathbf{x}_*) \geq f(\mathbf{x})$.

The three constraints are incorporated into $p(f(\mathbf{x})|D_n)$ by multiplying it with factors that embodies these constraints. The approximation procedures are described briefly below and more detailed derivations can be found in [13].

First, Lobato et al. define two random variables $\mathbf{c} = \big[\mathbf{y}_n; \bigtriangledown f(\mathbf{x}_*); \text{upper}[\bigtriangledown^2 f(\mathbf{x}_*)]\big] = [\mathbf{y}_n; \mathbf{0}; \mathbf{0}]$ and $\mathbf{z} = \big[f(\mathbf{x}_*); \text{diag}[\bigtriangledown^2 f(\mathbf{x}_*)]\big]$. The constraint C 1.1 is enforced by conditioning on $\mathbf{c}$:

$$p(\mathbf{z}|D_n, C1.1) = p(\mathbf{z}|\mathbf{c}) = \mathcal{N}(\mathbf{z}|\mathbf{m}_0, \mathbf{V}_0). \tag{2.22}$$

Constraints C 1.2 and C 2 are then included via the expression:

$$p(\mathbf{z}|D_n, C1, C2) \propto \Phi(f(\mathbf{x}_*) - y_{max}) \left[\prod_{i=1}^{d} \mathbb{I}([\bigtriangledown^2 f(\mathbf{x}_*)]_{ii} \leq 0)\right] \mathcal{N}(\mathbf{z}|\mathbf{m}_0, \mathbf{V}_0) \tag{2.23}$$

where the zero-mean Gaussian cumulative distribution $\Phi(f(\mathbf{x}_*) - y_{max})$ enforces constraint C2 and the product of d indicator functions encodes constraint C1.2. To facilitate the computation of the integral (Equation 2.25) later, Expectation Propagation (EP) technique is employed to approximate the non-Gaussian factors in Equation 2.23 with Gaussian distributions [4]. The EP approximation gives:

$$p(\mathbf{z}|D_n, C1, C2) \approx q(\mathbf{z}) \propto \left[\prod_{i=1}^{d+1} \mathcal{N}(z_i|\tilde{m}_i, \tilde{v}_i)\right] \mathcal{N}(\mathbf{z}|\mathbf{m}_0, \mathbf{V}_0). \tag{2.24}$$

Now define a new random variable $\mathbf{f} = [f(\mathbf{x}); f(\mathbf{x}_*)]$. The posterior distribution given observation data D_n and the constraints C1 and C2 can be approximated as

$$\begin{aligned} p(\mathbf{f}|D_n, C1, C2) &= \int p(\mathbf{f}|\mathbf{z}, D_n, C1, C2) p(\mathbf{z}|D_n, C1, C2) \mathrm{d}\mathbf{z} && (2.25)\\ &\approx \int p(\mathbf{f}|\mathbf{z}, D_n, C1, C2) q(\mathbf{z}) \mathrm{d}\mathbf{z} && (2.26)\\ &= \mathcal{N}(\mathbf{f}; \mathbf{m_f}, \mathbf{V_f}). && (2.27) \end{aligned}$$

Finally the constraint C3 can be incorporated in the following manner:

$$p(f(\mathbf{x})|D_n, C1, C2, C3) \approx Z^{-1} \int \mathbb{I}\big(f(\mathbf{x}) < f(\mathbf{x}_*)\big) \mathcal{N}(\mathbf{f}; \mathbf{m_f}, \mathbf{V_f}) \mathrm{d}f(\mathbf{x}_*) \tag{2.28}$$

where Z is a normalisation constant.

The variance of the final posterior distribution conditioned on all three constraints (Equation 2.28) is given by:

$$v_n(\mathbf{x}|\mathbf{x}_*) = [\mathbf{V_f}]_{1,1} - v^{-1}\beta(\beta+\alpha)\left\{[\mathbf{V_f}]_{1,1} - [\mathbf{V_f}]_{1,2}\right\}^2 \tag{2.29}$$

where $v = [-1, 1]^T \mathbf{V_f}[-1, 1]$, $\alpha = m/\sqrt{v}$, $m = [-1, 1]^T \mathbf{m_f}$, $\beta = \phi(\alpha)/\Phi(\alpha)$ and $\phi(\cdot)$ and $\Phi(\cdot)$ are standard normal pdf and cdf.

With the expression of the variance (Equation 2.29), we can obtain a closed-form approximation for the entropy of the predictive posterior distribution $p(y|D_n, \mathbf{x}, \mathbf{x}_*)$:

$$H[p(y|D_n, \mathbf{x}, \mathbf{x}_*)] \approx 0.5 \log\left[2\pi e\big(v_n(\mathbf{x}|\mathbf{x}_*) + \sigma_n^2\big)\right]. \tag{2.30}$$

Therefore, PES acquisition function (Equation 2.18) can be rewritten as:

$$\alpha_{PES}(\mathbf{x}|D_n) = 0.5 \log\left[2\pi e\big(v_n(\mathbf{x}) + \sigma_n^2\big)\right] - \frac{1}{N}\sum_{i=1}^{N} 0.5 \log\left[2\pi e\big(v_n(\mathbf{x}|\mathbf{x}_*^{(i)}) + \sigma_n^2\big)\right] \tag{2.31}$$

On top of the above approximations, PES treats the hyperparameters $\boldsymbol{\theta}$ in a fully Bayesian approach whereby the acquisition function (Equation 2.31) is marginalised over hyperparameters [13]. The intractable marginalisation procedure is approximated by the Monte Carlo integral [16]. By sampling

M sets of hyperparameters $\{\boldsymbol{\theta}^{(j)}|j = 1,\ldots,M\}$ from their posterior distribution $p(\boldsymbol{\theta}|D_n)$ [17], the resultant PES acquisition function becomes:

$$\begin{aligned}\alpha_{PES}(\mathbf{x}|D_n) &= \frac{1}{M}\sum_{j=1}^{M}\left\{0.5\log[2\pi e(v_n(\mathbf{x}|\boldsymbol{\theta}^{(j)}) + \sigma_n^2)] - \frac{1}{N}\sum_{i=1}^{N}0.5\log[2\pi e(v_n(\mathbf{x}|\mathbf{x}_*^{(i)},\boldsymbol{\theta}^{(j)}) + \sigma_n^2)]\right\}\\ &\approx \frac{1}{M}\sum_{j=1}^{M}\left\{0.5\log[2\pi e(v_n(\mathbf{x}|\boldsymbol{\theta}^{(j)}) + \sigma_n^2)] - 0.5\log[2\pi e(v_n(\mathbf{x}|\mathbf{x}_*^{(j)},\boldsymbol{\theta}^{(j)}) + \sigma_n^2)]\right\}\\ &\approx \frac{1}{M}\sum_{j=1}^{M}\left\{0.5\log[(v_n(\mathbf{x}|\boldsymbol{\theta}^{(j)}) + \sigma_n^2)] - 0.5\log[(v_n(\mathbf{x}|\mathbf{x}_*^{(j)},\boldsymbol{\theta}^{(j)}) + \sigma_n^2)]\right\}\end{aligned} \tag{2.32}$$

In conclussion, the PES algorithm can be summarised as follows:

Algorithm 2 PES Method

Input: a candidate $\mathbf{x}$; observation data $D_n = \{(\mathbf{x_i}, y_i)|i = 1,\ldots,M\}$

```
sample $M$ hyperparameter values $\theta^{(i)}$
for i=1,..., $M$ do
    use $f^{(i)}(\mathbf{x}) = \boldsymbol{\phi}^{(i)}(\mathbf{x})^T\boldsymbol{\psi}^{(i)}$ to approximate $p(f|D_n,\boldsymbol{\phi},\theta^{(i)}) = \mathcal{GP}(m_f(\cdot),K_f(\cdot,\cdot))$
    set $\mathbf{x}_*{}^{(i)} \leftarrow \arg\min_{\mathbf{x}\in\mathcal{X}} f^{(i)}(\mathbf{x})$
    approximate $p(f^{(i)}|D_n,\theta^{(i)},\mathbf{x},\mathbf{x}_*)$ with $p(f^{(i)}|D_n,\theta^{(i)},C1,C2,C3)$
    compute $\mathbf{m_0}{}^{(i)},\mathbf{V_0}{}^{(i)}$ for $p(f^{(i)}|D_n,\theta^{(i)},C1.1)$
    compute $\mathbf{m_f}{}^{(i)},\mathbf{V_f}{}^{(i)}$ for $p(f^{(i)}|D_n,\theta^{(i)},C1,C2)$ using Expectation Propogation(EP)
    compute $p(f^{(i)}|D_n,\theta^{(i)},C1,C2,C3) \approx Z^{-1}\int \mathbb{I}\big(f(\mathbf{x}) \geq f(\mathbf{x}_*)\big)\mathcal{N}(\mathbf{m_f}{}^{(i)},\mathbf{V_f}{}^{(i)})df(\mathbf{x}_*)$
    compute $v_n^{(i)}(\mathbf{x}|\mathbf{x}_*^{(i)})$ which is the variance of $p(f^{(i)}|D_n,\theta^{(i)},C1,C2,C3)$
    compute $v_n^{(i)}(\mathbf{x})$ directly
end for
return $\alpha_{PES}(\mathbf{x}|D_n) = \frac{1}{M}\sum_{i=1}^{M}\{0.5\log[v_n^{(i)}(\mathbf{x}) + \sigma_n^2] - 0.5\log[v_n^{(i)}(\mathbf{x}|\mathbf{x}_*^{(i)}) + \sigma_n^2]\}$
```

2.1.3.2 Limitations of PES

This section focuses on the limitations of the PES method and discusses how the ESBOPA method can overcome some, if not all, of them.

Covariance Limitations

The implementation of PES imposes several restrictions on the choice of appropriate covariance functions $k(\mathbf{x},\mathbf{x}')$ [?]. First, one key constraint used in approximating the analytically intractable posterior distribution $p(f(\mathbf{x})|D_n,\mathbf{x},\mathbf{x}_*)$, is the local maximum constraint (C 1) (i.e. $\nabla f(\mathbf{x}_*) = \mathbf{0}$ and $\nabla^2 f(\mathbf{x}_*)$ is negative definite) [13]. To incorporate this constraint, we must be able to compute the first and second partial derivatives of the covariance function ∇k and $\nabla^2 k$ [?]. This limits our choices of covariance functions to the ones whose derivatives have analytical expressions (e.g. squared exponential covariance) or are easy to approximate. Second, the approximation of the posterior distribution $p(\mathbf{x}_*|D_n)$, from which the global maximiser $\mathbf{x}_*$ is sampled, depends heavily on Bochner's theorem and thus demands the use of stationary covariances. Moreover, the construction of the feature vector $\phi(\mathbf{x})$ in approximating $p(\mathbf{x}_*|D_n)$ requires sampling from the normalised spectral density of the covariance function. Hence, any stationary covariances whose spectral density is difficult for us to sample from may not be suitable for PES method. Given the critical role of the covariance function in the Gaussian process, the restricted covariance choices faced by PES method will limit its applicability and performance in solving real-world problems that require the use of complex or even non-stationary covariance function.

ESBOPA method does not suffer such covariance constraints because it doesn't require the derivatives or spectral density of the covariance function. Furthermore, ESBOPA method can be derived without using the Bochner's theorem and thus can be extended to real world applications that requires non-stationary covariances.

Input-space constraints

Information-theoretic methods mentioned in Section 2.1.2.3 , such as ES and PES, all aim to maximise the information gain (or entropy reduction) about the latent maximiser $\mathbf{x}_*$ and thus deal with the input space [18]. This reduces the efficiency of PES approach in high dimensional problems where a large number of maximiser samples from the input space are needed [18] and each sample is computationally expensive to obtain.

In view of these problems, more recent information-theoretic methods such as Output-space Predictive Entropy Search (OPES) [Hoffman and Ghahramani, 2015] and Max-value Entropy Search (MES) [18] are developed by using the information content about the maximum function value $f(\mathbf{x}_*)$ instead of the maximiser $\mathbf{x}_*$. As the output space is always one-dimensional, these new methods enjoy a significantly lower computational burden in approximating mutual information via sampling [18] and can be applied to problems with a wider range of input space types [?, ?].

ESBOPA method also works in the output space rather than the input space. Thus, it possesses the same advantages as OPES and MES over the information-theoretic methods that work on inputs.

Additional sampling process

Current entropy search methods, being it dealing with maximiser or maximum value, all involve two sampling processes : 1) sampling hyperparameters for marginalisation and 2) sampling global optimum for entropy computation. The second sampling process is more complicated because it requires the construction of a good approximation for the objective function[13] which introduces some covariance restrictions.

In ESBOPA method, the parabolic approximation for the objective function $f(\mathbf{x}) = \eta + {}^{1}\!/\!{}_{2}g(\mathbf{x})^2$ allows us to explicitly express the minimum value as a hyperparameter η, thus circumventing the need for the second sampling process.

Tedious approximation process for the intractable conditional distribution

In PES, the intractable distribution $p(f|D, \mathbf{x}, \mathbf{x}_*)$ is approximated by conditioning $p(f|D)$ on three constraints to ensure that the maximum sampled is lower than observation data as well as any potential test points [13]. However, incorporating these 3 constraints, as illustrated in Section 2.1.3.1 incurs much difficulty as it involves the use of computationally expensive technique, Expectation Propagation (EP), as well as requires the covariance derivatives as mentioned above.

ESBOPA methods circumvent this difficult and tedious process by the use of parabolic approximation $f(\mathbf{x}) = \eta + {}^{1}\!/\!{}_{2}g(\mathbf{x})^2$. This parabolic form ensures that η is constrained to be a global minimum, thus implicitly fulfilling the 3 constraints used in PES.

3 Entropy Search for Bayesian Optimisation Based on Parabolic Approximation

3.1 Parabolic Approximation and Linearisation

The idea of parabolic approximation is inspired by [19] where similar transformation is used to preserve non-negativity of the likelihood functions and deal with the high dynamic range of likelihood values. The parabolic approximation for the unknown objective function plays a central role in our approach of entropy search for Bayesian optimisation based on parabolic approximation(ESBOPA). It enables us to explicitly capture the global minimum as a hyperparameter η, thus circumventing the additional sampling process for global minimum as well as the procedures to constrain the sample minimum to be the global minimum. This section explains the parabolic approximation in detail and how we can harness a linearisation technique to extend the benefits of the parabolic approximation.

An unknown objective function $f(\mathbf{x})$ can be expressed in the parabolic form [19]:

$$f(\mathbf{x}) = \eta + {}^1\!/\!{}_2 g(\mathbf{x})^2 \tag{3.1}$$

where η is the global minimum of the objective function. Given the noise-free observation data $D_f = \{(\mathbf{x_i}, f_i)|i = 1, \ldots n\} = \{\mathbf{X}_n, \mathbf{f}_n\}$, the observation data on g is $D_g = \{(\mathbf{x_i}, g_i)|i = 1, \ldots n\} = \{\mathbf{X}_n, \mathbf{g}_n\}$ where $g_i = \sqrt{2(f_i - \eta)}$.

We choose a zero-mean Gaussian process prior on $g(\mathbf{x})$: $g \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}'))$ so that the posterior distribution for g given the observation data D_g and the test point $\mathbf{x}$ is also a Gaussian process:

$$p(g|D_g, \mathbf{x}, \eta) = \mathcal{GP}\big(g; m_g(\cdot), K_g(\cdot, \cdot)\big) \tag{3.2}$$

where

$$m_g(\mathbf{x}) = K(\mathbf{x}, \mathbf{X}_n)K(\mathbf{X}_n, \mathbf{X}_n)^{-1}\mathbf{g}_n \quad \text{and} \tag{3.3}$$

$$K_g(\mathbf{x}, \mathbf{x}') = K(\mathbf{x}, \mathbf{x}') - K(\mathbf{x}, \mathbf{X}_n)K(\mathbf{X}_n, \mathbf{X}_n)^{-1}K(\mathbf{X}_n, \mathbf{x}). \tag{3.4}$$

However, due to the parabolic transformation, the distribution for any f is now a non-central χ^2 distribution, which would make the analysis intractable. In order to tackle this problem and obtain a posterior distribution $p(f|D_f, \mathbf{x}, \eta)$ that is also Gaussian, we need to resort to the linearisation technique proposed in [19].

The technique performs a local linearisation of the parabolic transformation $h : f = \eta + {}^1\!/\!{}_2 g^2$. By linearising around g_0, we obtain the expression $f \approx h(g_0) + h'(g_0)(g - g_0)$ where the gradient $h'(g) = g$. This neat gradient form is another advantage of using the parabolic transformation. We set $g_0 = m_g$ which represents the mode of the posterior distribution $p(g|D_g, \mathbf{x}, \eta)$ and then get an expression for f which is linear in g:

$$f(\mathbf{x}) \approx [\eta + {}^1\!/\!{}_2 m_g(\mathbf{x})^2] + m_g(\mathbf{x})[g(\mathbf{x}) - m_g(\mathbf{x})] = \eta - {}^1\!/\!{}_2 m_g(\mathbf{x})^2 + m_g(\mathbf{x})g(\mathbf{x}). \tag{3.5}$$

Since Gaussian processes are closed under affine transformations, the predictive posterior distribution for f conditioned on the noise-free observation data D_f and a test point $\mathbf{x}$ also remains as a Gaussian process:

$$p(f|D_f, \mathbf{x}, \eta) = \mathcal{GP}\big(f; m_f(\cdot), K_f(\cdot, \cdot)\big) \tag{3.6}$$

where

$$m_f(\mathbf{x}) = \eta + {}^1\!/\!{}_2 m_g(\mathbf{x})^2 \tag{3.7}$$

$$K_f(\mathbf{x}, \mathbf{x}') = m_g(\mathbf{x})K_g(\mathbf{x}, \mathbf{x}')m_g(\mathbf{x}'). \tag{3.8}$$

However, in real world situation, we do not have access to the true function values but only noisy observation of the function $y(\mathbf{x}) = f(\mathbf{x}) + \epsilon$ where ϵ is assumed to be an independently and identically distributed Gaussian noise with variance σ_n^2 [4]. In the case of noisy observation data $D_n = \{(\mathbf{x_i}, y_i)|i = 1, \dots n\} = \{\mathbf{X}_n, \mathbf{y}_n\}$, the posterior distribution for y is also a Gaussian process because:

$$p(y|D_n, \mathbf{x}, \eta) = \int p(y|f)p(f|D_n, \mathbf{x}, \eta)df = \int \mathcal{N}(y; f, \sigma_n^2)\mathcal{N}\big(f; m_f(\cdot), K_f(\cdot, \cdot)\big)\mathrm{d}f \tag{3.9}$$

and by using the properties of Gaussians [5], we arrive at:

$$p(y|D_n, \mathbf{x}, \eta^{(i)}) = \mathcal{GP}\big(y; m_f(\cdot), K_f(\cdot, \cdot) + \sigma_n^2\big) \tag{3.10}$$

where $m_f(\cdot)$ and $K_f(\cdot, \cdot)$ have the same form as Equation 3.7 and 3.8.

3.2 Derivation of ESBOPA Algorithm

This section explains the detailed derivation of ESBOPA acquisition function. With reference to Section 2.1.2.3, the acquisition function for ES is

$$\alpha_{ES}(\mathbf{x}|D_n) = H[p(\mathbf{x}_*|D_n)] - \mathbb{E}_{p(y|D_n, \mathbf{x})}\Big[H\big[p\big(\mathbf{x}_*|D_n \cup (\mathbf{x}, y)\big)\big]\Big] \tag{3.11}$$

PES makes use of the symmetry of mutual information and arrives at the following equivalent acquisition function:

$$\alpha_{PES}(\mathbf{x}|D_n) = H[p(y|D_n, \mathbf{x})] - \mathbb{E}_{p(\mathbf{x}_*|D_n)}\Big[H\big[p(y|D_n, \mathbf{x}, \mathbf{x}_*)\big]\Big] \tag{3.12}$$

where $p(y|D_n, \mathbf{x}, \mathbf{x}_*)$ is the predictive posterior distribution for y conditioned on the observed data D_n, the test location $\mathbf{x}$ and the global minimiser $\mathbf{x}_*$ of the objective function.

ESBOPA uses the same information-theoretic thinking but measures the entropy about the latent minimum value $f_* = f(\mathbf{x}_*)$ instead of that of the latent minimiser $\mathbf{x}_*$. Thus, the acquisition function of the ESBOPA method is the mutual information between the function minimum f_* and the next query point [18]. In other words, ESBOPA aims to select the next query point which minimise the entropy of the latent minimum value:

$$\alpha_{ESBOPA}(\mathbf{x}|D_n) = H[p(y|D_n, \mathbf{x})] - \mathbb{E}_{p(f_*|D_n)}\Big[H\big[p(y|D, \mathbf{x}, f_*)\big]\Big]. \tag{3.13}$$

We express the unknown objective function in a parabolic form: $f(\mathbf{x}) = \eta + {}^1\!/\!_2 g(\mathbf{x})^2$.Thus,$f_* = \eta$ in our case and ESBOPA acquisition function can be reformulated as:

$$\alpha_{ESBOPA}(\mathbf{x}|D_n) = H[p(y|D_n, \mathbf{x})] - \mathbb{E}_{p(\eta|D_n)}\Big[H\big[p(y|D, \mathbf{x}, \eta)\big]\Big] \tag{3.14}$$

where $p(\eta|D)$ is the posterior distribution of η.

This idea of changing entropy computation from the input space to the output space is also shared by [**?**] and [18]. Hence, the acquisition function of the ESBOPA method is very similar to those of OPES[**?**] and MES [18]. Making such change also brings additional benefits such as wider applicability and lower computational costs as discussed in Section 2.1.3.2.

The acquisition function of ESBOPA method can then be manipulated as follows:

$$\begin{aligned}\alpha_{ESBOPA}(\mathbf{x}|D_n) &= H[p(y|D_n, \mathbf{x})] - \mathbb{E}_{p(\eta|D_n)}\Big[H\big[p(y|D_n, \mathbf{x}, \eta)\big]\Big] \\ &= H\Big[\int p(y|D_n, \mathbf{x}, \eta)p(\eta|D_n)\mathrm{d}\eta\Big] - \int p(\eta|D_n)H\big[p(y|D_n, \mathbf{x}, \eta)\big]\mathrm{d}\eta.\end{aligned}$$

The integral terms in the above expression can be estimated via the Monte Carlo integration[5]:

$$\int f(z)p(z)\mathrm{d}z = \mathbb{E}_p[f(z)] \approx \frac{1}{N}\sum_i^N f(z^{(i)}). \tag{3.15}$$

By drawing N samples of η from the posterior distribution $p(\eta|D)$ and using the Monte Carlo approximation, ESBOPA acquisition function can be rewritten as

$$\alpha_{ESBOPA}(\mathbf{x}|D_n) \approx H\Big[\frac{1}{N}\sum_i^N p(y|D_n, \mathbf{x}, \eta^{(i)})\Big] - \frac{1}{N}\sum_i^N H[p(y|D_n, \mathbf{x}, \eta^{(i)})]. \tag{3.16}$$

The expression 3.16 can be viewed as the difference between the entropy of the expectation of a variable and the expected entropy of the variable

$$\alpha_{ESBOPA}(\mathbf{x}|D_n) = H\big[\mathbb{E}[Z]\big] - \mathbb{E}\big[H[Z]\big]. \tag{3.17}$$

Provided that differential entropy $H[Z]$ is a concave function, by Jensen's inequality,

$$H\big[\mathbb{E}[Y]\big] \geq \mathbb{E}\big[H[Z]\big]. \tag{3.18}$$

Therefore, our ESBOPA acquisition function will definitely give a non-negative mutual information/information gain.

With reference to Equation 3.16, the three crucial components in computing the ESBOPA acquisition function are 1) the posterior distribution $p(\eta|D_n)$ from which we draw η samples, 2) the first term which is the entropy of a Gaussian mixture and 3) the second term which is expected entropy of a Gaussian. The following subsections will explain how we compute these three components.

3.2.1 Entropy of a Gaussian Mixture

The first term in the ESBOPA acquisition function is the entropy of a gaussian mixture which does not have a closed form. To approximate this entropy term, we need to use one of the approximation techniques discussed in Section 3.5.

3.2.2 Posterior Distribution on $\boldsymbol{\eta}$

By Bayes rule,

$$p(\eta|D_n) = \frac{p(D_n|\eta)p(\eta)}{\int p(D_n|\eta)p(\eta)d\eta} \propto p(D_n|\eta)p(\eta)$$

Let $\tilde{p}(\eta|D_n) = p(D_n|\eta)p(\eta)$ and take the log of the unnormalised posterior distribution

$$\log \tilde{p}(\eta|D_n) = \log p(D_n|\eta) + \log p(\eta) \tag{3.19}$$

where $\log p(D_n|\eta) = -\frac{1}{2}\mathbf{y}_n^T[K_f^\eta(\mathbf{X}_n, \mathbf{X}_n') + \sigma_n^2\mathbf{I}]^{-1}\mathbf{y}_n - \frac{1}{2}\log|K_f^\eta(\mathbf{X}_n, \mathbf{X}_n') + \sigma_n^2\mathbf{I}| - \frac{n}{2}\log 2\pi$ [4].

The superscript η denotes that the covariance matrix K_f^η depends on η as illustrated in section 3.1. Thus, η can be simply treated as a hyperparameter and can be sampled together with other hyperparameters via any MCMC sampling algorithm (e.g. Metropolis-Hastings algorithm). The details of hyperparameter treatment would be discussed further in Section 3.3.

3.2.3 Expected Entropy of a Gaussian Process

The second term in ESBOPA acquisition function is the expected entropy of the predictive posterior distribution which is also a Gaussian process as illustrated in Section 3.1: $p(y|D_n, \mathbf{x}, \eta) =$

$\mathcal{GP}(y; m_f(\cdot), K_f(\cdot,\cdot) + \sigma_n^2)$ (Equation 3.10). The entropy of a Gaussian is an analytical function of its variance $v_f(\mathbf{x}|D_n, \eta) = K_f(\mathbf{x}, \mathbf{x}')$:

$$H[p(y|D_n, \mathbf{x}, \eta^{(i)})] = 0.5 \log \big[2\pi e \big(v_f(\mathbf{x}|D_n, \eta^{(i)}) + \sigma_n^2\big)\big].$$

Thus, the second term in the acquisition function (Equation 3.16) can be computed explicitly

$$\frac{1}{N}\sum_i^N H[p(y|D_n, \mathbf{x}, \eta^{(i)})] = \frac{1}{2N}\sum_i^N \log \big[2\pi e \big(v_f(\mathbf{x}|D_n, \eta^{(i)}) + \sigma_n^2\big)\big] \tag{3.20}$$

After solving all three components, we can rewrite the acquisition function (equation 3.16) as

$$\alpha_{ESBOPA}(\mathbf{x}|D_n) \approx H\Big[\frac{1}{N}\sum_i^N p(y|D_n, \mathbf{x}, \eta^{(i)})\Big] - \frac{1}{2N}\sum_i^N \log \big[2\pi e \big(v_f(\mathbf{x}|D_n, \eta^{(i)}) + \sigma_n^2\big)\big] \tag{3.21}$$

3.3 Hyperparameter Tunning

Hyperparameters are the free parameters, such as output scale, characteristic length scales and noise variance, in the covariance function. The above analysis of ESBOPA acquisition function does not consider the learning of hyperparameters but different hyperparameter values will heavily affect the inference performance of a Gaussian process model [4]. There are mainly two approaches to learn hyperparameters. First, we could learn the optimal set of hyperparameter values via maximum likelihood estimation(MLE) or maximum a posterior estimation (MAP). Second, we could marginalise over hyperparameters to perform inference [**?**].

3.3.1 Maximum Likelihood Estimation

The maximum likelihood estimation (MLE) approach, as the name suggested, is to learn the optimal set of the hyperparameter values $\boldsymbol{\theta}$ by maximising the log marginal likelihood function $\log p(D_n|\boldsymbol{\theta})$. The log marginal likelihood function for a zero-mean Gaussian Process model has the form

$$\log p(D_n|\boldsymbol{\theta}) = -\frac{1}{2}\mathbf{y}_n^T[K_{\mathbf{y}}(\mathbf{X}_n, \mathbf{X}_n')]^{-1}\mathbf{y}_n - \frac{1}{2}\log|K_{\mathbf{y}}(\mathbf{X}_n, \mathbf{X}_n')| - \frac{n}{2}\log 2\pi \tag{3.22}$$

where $K_{\mathbf{y}}(\mathbf{X}_n, \mathbf{X}_n') = K_{\mathbf{f}}(\mathbf{X}_n, \mathbf{X}_n') + \sigma_n^2\mathbf{I}$ is the covariance matrix for the noisy observation data $D_n = \{\mathbf{X}_n, \mathbf{y}_n\}$, which also depends on $\boldsymbol{\theta}$.

With reference to Equation 3.22, the log marginal likelihood comprises three terms: the first term $-0.5\mathbf{y}_n^T[K_{\mathbf{y}}(\mathbf{X}_n, \mathbf{X}_n')]^{-1}\mathbf{y}_n$ measures how well the Gaussian Process model fits the data; the second term $-0.5\log|K_{\mathbf{y}}(\mathbf{X}_n, \mathbf{X}_n')|$ penalises excessive model complexity; the last term $0.5n\log 2\pi$ is a normalising constant [4]. The effect of the first two terms are illustrated in Figure **??**.The model fit term decreases as the length scale rises because the model becomes less and less flexible. On the other hand, the term of negative complexity penalty grows with the length scale as the model complexity decreases with larger length scale. Therefore, the maximum likelihood estimation inherently balances the trade-off between model fit and model complexity.

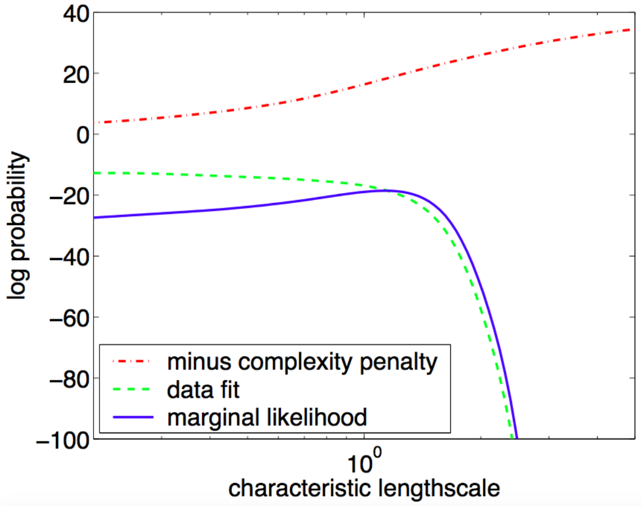

Figure 3.1: *The log marginal likelihood function and the effect of its components[4]*

To facilitate the optimisation, we can compute the gradients of the log marginal likelihood function with respect to hyperparameters, which has the form

$$\frac{\partial \log p(D|\boldsymbol{\theta})}{\partial \theta_i} = \frac{1}{2}\mathbf{y}_n^T K_{\mathbf{y}}^{-1}\frac{\partial K_{\mathbf{y}}}{\partial \theta_i}K_{\mathbf{y}}^{-1}\mathbf{y}_n - \frac{1}{2}\mathrm{tr}\left(K_{\mathbf{y}}^{-1}\frac{\partial K_{\mathbf{y}}}{\partial \theta_i}\right) = \frac{1}{2}\mathrm{tr}\left[(\boldsymbol{\alpha}\boldsymbol{\alpha}^T - K_{\mathbf{y}}^{-1})\frac{\partial K_{\mathbf{y}}}{\partial \theta_i}\right] \tag{3.23}$$

where $\boldsymbol{\alpha} = K_{\mathbf{y}}^{-1}\mathbf{y}_n$. Then, effective gradient-based optimisation algorithms such as conjugate gradients [5] can be employed to find the optimal set of hyperparameters.

However, the log marginal likelihood function is generally a non-convex function with multiple local optima [5]. Thus, maximum likelihood estimation may lead to hyperparameters corresponding to a bad local optimum, causing the problem of over-fitting. This is more likely to occur when the observation/training data set is small, which is the case for Bayesian optimisation, and the local optima of marginal likelihood are quite close [4].

If we have some prior knowledge about hyperparameters, we can combine the prior distribution over $\boldsymbol{\theta}$ with the marginal likelihood to implement the maximum a posteriori estimation. By Bayes rule, the posterior over the hyperparameters is

$$p(\boldsymbol{\theta}|D_n) = \frac{p(D_n|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int p(D_n|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}} \propto p(D_n|\boldsymbol{\theta})p(\boldsymbol{\theta}) \tag{3.24}$$

$$\Rightarrow \log p(\boldsymbol{\theta}|D_n) \propto \log p(D_n|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta}). \tag{3.25}$$

Thus, maximising the log posterior over hyperparameters is equivalent to maximising the sum of the log marginal likelihood and the log prior. The maximum a posteriori estimation can be viewed as the maximum likelihood estimation with regulation, thus alleviating the problem of overfitting.

3.3.2 Marginalisation

Both maximum likelihood estimation and maximum a posteriori estimation are not desirable as they give point estimates and ignore our uncertainty about the hyperparameters. In a fully Bayesian treatment of the hyperparameters, we should consider all possible hyperparameter values by marginalising the acquisition function (Equation 3.21) with respect to the posterior $p(\boldsymbol{\psi}|D_n)$ where $\boldsymbol{\psi} = \{\boldsymbol{\theta}, \eta\}$:

$$\alpha_{ESBOPA}(\mathbf{x}|D_n) = \int \alpha_{ESBOPA}(\mathbf{x}|\boldsymbol{\Psi}, D_n) p(\boldsymbol{\Psi}|D_n) d\boldsymbol{\Psi}. \tag{3.26}$$

Since complete marginalisation over hyperparameters is analytically intractable, the integral must be approximated using the Monte Carlo method [?] [16]:

$$\alpha_{ESBOPA}(\mathbf{x}|D_n) = H\Big[\frac{1}{N}\sum_i^N p(y|D_n, \mathbf{x}, \boldsymbol{\theta}^{(i)}, \eta^{(i)})\Big] - \frac{1}{2N}\sum_i^N \log\big[2\pi e\big(v_f(\mathbf{x}|D, \boldsymbol{\theta}^{(i)}, \eta^{(i)}) + \sigma_n^2\big)\big] \tag{3.27}$$

3.4 ESBOPA Algorithm

The procedures of ESBOPA approach can be summarised by the following algorithm

Algorithm 3 ESBOPA Version2

```
Input: a test input $\mathbf{x}$; noisy observation data $D_n = \{(\mathbf{x}_i, y_i)|i = 1, \dots, n\}$

sample hyperparameters and $\eta$ : $\boldsymbol{\Psi} = \{\boldsymbol{\Theta}, \boldsymbol{\eta}\} = \{\boldsymbol{\theta}^{(j)}, \eta^{(j)}|j = 1, \dots, M\}$ from the posterior distribution $p(\boldsymbol{\psi}|D_n)$
for j=1,..., $M$ do
    use $f(\mathbf{x}) = \eta + 1/2 g(\mathbf{x})^2$ to approximate $p(f|D_n, \mathbf{x}, \boldsymbol{\theta}^{(j)}, \eta^{(j)}) = \mathcal{GP}\big(m_f(\cdot), K_f(\cdot,\cdot)\big)$
      3.1) compute $D_g = \{(\mathbf{x}_i, g_i)|i = 1, \dots, n\}$ where $g_i = \sqrt{2(y_i - \eta^{(j)})}$
      3.2) compute posterior distribution $p(g|D_g, \mathbf{x}, \boldsymbol{\theta}^{(j)}, \eta^{(j)}) = \mathcal{GP}\big(g; m_g(\cdot), K_g(\cdot,\cdot)\big)$
      3.3) approximate $p(f|D_n, \mathbf{x}, \boldsymbol{\theta}^{(j)}, \eta^{(j)})$ using linearisation technique
    compute $p(y|D_n, \mathbf{x}, \boldsymbol{\theta}^{(j)}, \eta^{(j)}) = \mathcal{GP}\big(m_f(\cdot), K_f(\cdot,\cdot) + \sigma_n^2\big)$
    compute $H[p(y|D_n, \mathbf{x}, \boldsymbol{\theta}^{(j)}, \eta^{(j)})] = 0.5 \log\big[2\pi e\big(v_f(\mathbf{x}|D_n, \boldsymbol{\theta}^{(j)}, \eta^{(j)}) + \sigma_n^2\big)\big]$
end for
estimate entropy of the gaussian mixture :
      $E1(\mathbf{x}|D_n) = H\Big[\frac{1}{M}\sum_j^M p(y|D_n, \mathbf{x}, \boldsymbol{\theta}^{(j)}, \eta^{(j)})\Big]$
compute the entropy expectation: $E2(\mathbf{x}|D_n) = \frac{1}{2M}\sum_j^M \log\big[2\pi e\big(v_f(\mathbf{x}|D_n, \boldsymbol{\theta}^{(j)}, \eta^{(j)}) + \sigma_n^2\big)\big]$
return $\alpha_n(\mathbf{x}|D_n) = E1(\mathbf{x}|D_n) - E2(\mathbf{x}|D_n)$
```

Figure **??** illustrates the sampling behaviour of ESBOPA method for a simple 1-D Bayesian optimisation problem. The optimisation process is started with three initial observation data. Subsequent query points are selected by maximising ESBOPA acquisition function which trades off exploitation and exploration. As more samples are taken, the mean of the posterior distribution for the objective function gradually resembles the objective function $f(\mathbf{x})$. The distribution for η is set such that η value is always lower than the minimum observation. As more information is gained, the distribution of η converges to the global minimum $f(\mathbf{x}_*)$.

3.5 Approximation of a Gaussian Mixture Entropy

3.5.1 Method 1: Taylor Expansion

Huber et al. (2008) [20] propose a novel method for approximating the entropy of a Gaussian mixture model by using a Taylor-series expansion of the logarithm of the Gaussian mixture.

Let $q(y) = \sum_i^N w_i p(y|D_n, \mathbf{x}, \eta^{(i)}) = \sum_i^N w_i \mathcal{N}(y; m_i, \sigma_i^2)$. The Gaussians in the mixture are univariate in our case because the function value at a test location $\mathbf{x}$ is 1-D. The entropy of this mixture is:

$$H\left[\frac{1}{N}\sum_i^N p(y|D, \mathbf{x}, \eta^{(i)})\right] = H\left[q(y)\right] = -\int q(y) \log h(y) \mathrm{d}y \tag{3.28}$$

where $h(y) = q(y)$ but we uses different notations to differentiate the Gaussian mixture that's argument of the logarithm from that in front of the logarithm.

By expanding the logarithm term around the mean of each Gaussian term m_i in $h(y)$, the resultant R-th order Taylor series is

$$\log h(y) = \sum_{k=0}^R \frac{(y-m_i)^k}{k!} \frac{\mathrm{d}^k\big(\log h(y)\big)}{\mathrm{d}y^k}\Big|_{y=m_i}. \tag{3.29}$$

We then substitute equation 3.29 into equation 3.28 and obtain

$$\begin{aligned}
H\left[q(y)\right] &= -\int q(y) \log h(y) dy \\
&= -\int \sum_i^N w_i \mathcal{N}(y; m_i, \sigma_i^2) \log h(y) \mathrm{d}y \\
&= -\sum_i^N w_i \int \mathcal{N}(y; m_i, \sigma_i^2) \sum_{k=0}^R \frac{(y-m_i)^k}{k!} \frac{\mathrm{d}^k\big(\log h(y)\big)}{\mathrm{d}y^k}\Big|_{y=m_i} \mathrm{d}y \\
&= -\sum_i^N w_i \sum_{k=0}^R \frac{1}{k!} \frac{\mathrm{d}^k\big(\log h(y)\big)}{\mathrm{d}y^k}\Big|_{y=m_i} \int \mathcal{N}(y; m_i, \sigma_i^2)(y-m_i)^k \mathrm{d}y
\end{aligned}$$

where $\int \mathcal{N}(y; m_i, \sigma_i^2)(y-m_i)^k \mathrm{d}y$ is the k-th central moment of a Gaussian distribution and thus has a closed form. The k-th derivative of the logarithm of Gaussian mixture $h(y)$ can also be computed

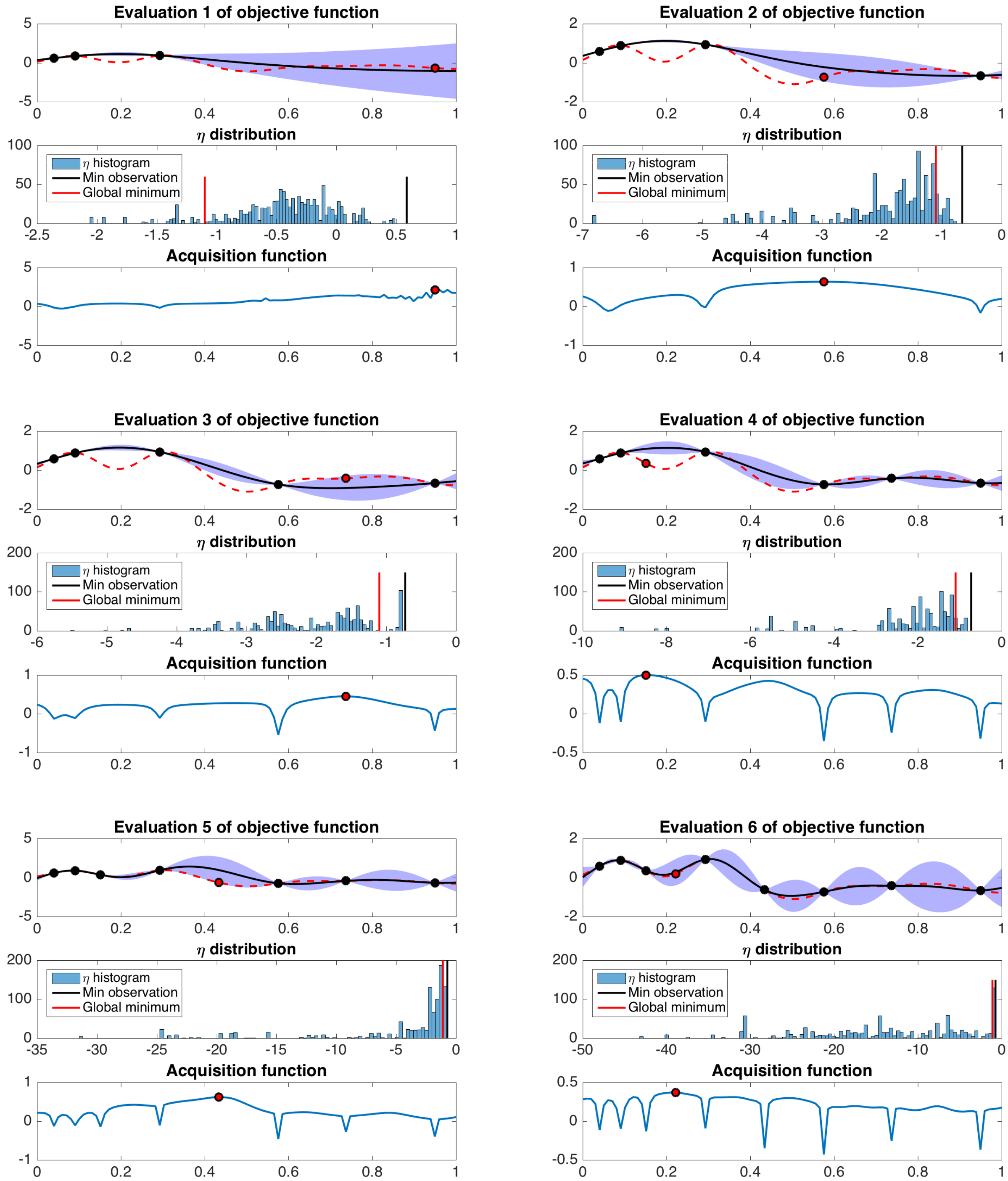

Figure 3.2: *Bayesian optimisation for a 1-D objective function using ESBOPA method. In each subfigure, the top plot shows the latent objective function (red dotted line), the posterior mean (black solid line) and the 95% confidence interval (blue shaded area) estimated by the Gaussian process model as well as the observation points (black solid dot) and the next query point (red solid dot with black edge).The middle plot is the histogram of η samples and its relation to the minimum observation (black vertical line) and the true global minimum (red vertical line). The bottom plot illustrates ESBOPA acquisition function which is maximised to select the next query point.*

analytically because the derivatives of a Gaussian distribution always exist and Kronecker algebra can be used to achieve a compact representation [20].

The entropy approximation by Taylor expansion faces the trade-off between the accuracy and computational burden as we can obtain more accurate approximations by including higher order Taylor-series terms at the expense of computational speed [20]. Experiments with this approximation approach are carried out with a second-order Taylor-series expansion whose explicit form is provided by the Appendix in [20]:

$$H\left[\frac{1}{N}\sum_{i=1}^{N} p(y|D_n,\mathbf{x},\eta^{(i)})\right] \approx H_0[y]+H_2[y] = -\sum_{i=1}^{N} w_i \log h(m_i) - \sum_{i=1}^{N}\frac{w_i\sigma_i^2}{2}F(m_i) \tag{3.30}$$

where $F(y) = h(y)^{-1}\sum_{j=1}^{N} w_j\sigma_j^{-2}\left[h(y)^{-1}(y-\mu_j)h'(y)+\sigma_j^{-2}(y-\mu_j)^2-1\right]\mathcal{N}(y;\mu_j,\sigma_j^2)$.

3.5.2 Method 2: Numerical Integration

As mentioned before, one advantage of ESBOPA method is that it allows us to transform the entropy calculation from the multi-dimensional input space to the one-dimensional output space. This, thus, permits the use of numerical integration techniques to effectively compute the entropy of a Gaussian mixture. Experiments with numerical integration are performed with the *quad* function in MATLAB which utilises the adaptive Simpson quadrature.

3.5.3 Method 3: Simple Monte Carlo

The first term in our ESBOPA acquisition function (Equation 3.21) can be reformulated in the following way:

$$\begin{aligned}
& H\left[\frac{1}{N}\sum_{i}^{N} p(y|D_n,\mathbf{x},\eta^{(i)})\right] \\
= \; & H\left[\sum_{i}^{N} w_i p(y|D_n,\mathbf{x},\eta^{(i)})\right] \qquad \text{where} \qquad w_i = \frac{1}{N} \\
= \; & -\int\left(\sum_{i}^{N} w_i p(y|D_n,\mathbf{x},\eta^{(i)})\right)\log\left(\sum_{i}^{N} w_i p(y|D_n,\mathbf{x},\eta^{(i)})\right)\mathrm{d}y \\
= \; & -\sum_{i}^{N} w_i \int p(y|D_n,\mathbf{x},\eta^{(i)})\log\left(\sum_{i}^{N} w_i p(y|D_n,\mathbf{x},\eta^{(i)})\right)\mathrm{d}y
\end{aligned}$$

By drawing M samples of y from $p(y|D_n,\mathbf{x},\eta^{(i)})$ and using Monte Carlo integration, the entropy of a Gaussian mixture can be approximated as

$$H\left[\frac{1}{N}\sum_{i}^{N} p(y|D_n,\mathbf{x},\eta^{(i)})\right] \approx -\sum_{i}^{N} w_i\left[\frac{1}{M}\sum_{j}^{M}\log\left(\sum_{i}^{N} w_i p(y^{(j)}|D_n,\mathbf{x},\eta^{(i)})\right)\right] \tag{3.31}$$

The accuracy of the simple Monte Carlo approximation can be enhanced by increasing the sample size M. But larger number of samples will increase the computational burden. Thus, we also face a trade-off between the approximation precision and computational speed.

3.5.4 Experiments for Comparing Approximation Methods

The following experiments are conducted to validate as well as compare the three entropy approximation methods: 1) Huber's method that uses Taylor series expansion (Huber), 2) numerical integration that uses adaptive Simpson quadrature (Quadra) and 3) the simple Monte Carlo integration (MC). The approximation performance is assessed in terms of accuracy and computational speed. The optimal approximation method is then chosen based on the trade-off between the accuracy and computational demand.

The methodology of the tests can be summarised as follows:

[1] Generate a Gaussian mixture as a weighted sum of N 1-D random Gaussian distributions

[2] For the Gaussian mixture, estimate its true entropy by using simple Monte Carlo method with large sample size (e.g. MC50000)

[3] Use the 3 approximation methods to approximate the entropy of the Gaussian mixture. For the MC method, try it with different sample sizes (e.g. MC10,MC100,MC1000)

[4] Compute and record the running time as well as absolute and fractional approximation errors for each method.

[5] Repeat the above processes for M different gaussian mixtures and compute the median running time and the median of the approximation errors.

With reference to Figure **??** and **??**, in the case of a single Gaussian($N = 1$) distribution, there is a closed-form expression for its entropy. The Huber's method gives the exact true entropy solution, thus having 0 approximation error. The other 2 approximation methods (Quadra and MC) are compared against the true entropy value. It is evident that the approximation by Monte Carlo with 50000 samples (MC50000) is very close to the true value, which justifies our use of the approximation results of MC50000 as our yardstick for the cases of more than one Gaussians in the mixture.

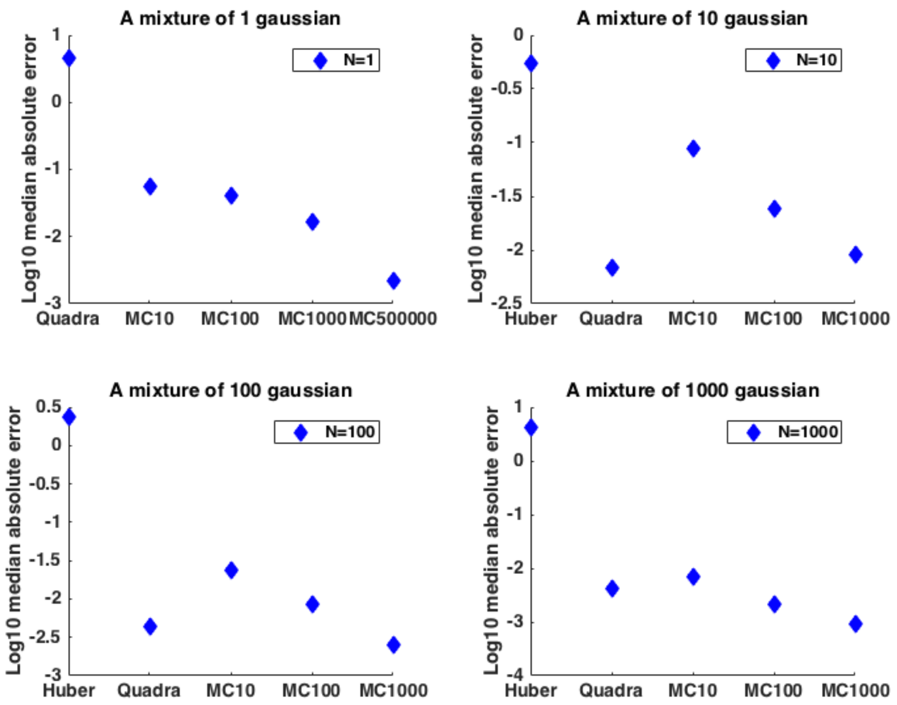

Figure 3.3: *Log median absolute error in approximating the entropy of a Gaussian mixture.*

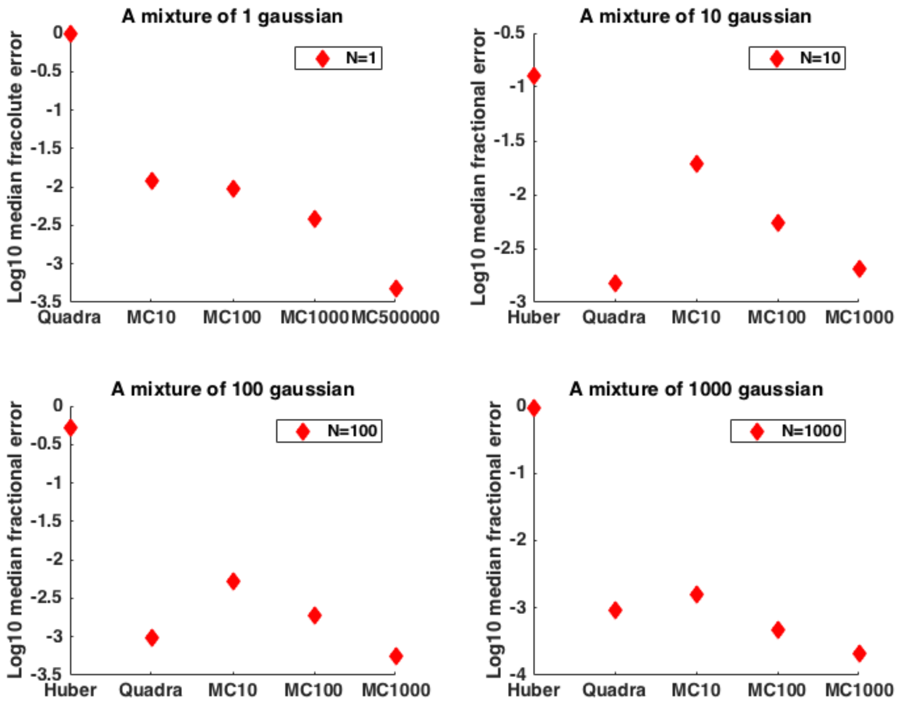

Figure 3.4: *Log median fractional error in approximating the entropy of a gaussian mixture*

For a mixture of more than one Gaussian distribution ($N > 1$), the performances of all 3 approximation methods (Huber, Quadra, MC) are compared against the entropy value estimated by MC50000. The results in Figure **??** and **??** show that Monte Carlo with a sample size of 1000 (MC1000) produces the most accurate approximation in terms of absolute and fractional approximation errors. MC100 and quadrature (Quadra) also have relatively accurate approximation with low absolute and fractional error. The Huber method leads to the highest approximation errors. This may be due to the low order (order of 2) of Taylor-series expansion used in our experiments.

In Figure **??**, the running times of all 3 approximation methods are compared. As expected, the results show that the computation time increases as the number of Gaussians in the mixture rises. This is mainly due to the computation burden associated with the construction of the Gaussian mixture. More importantly, the quadrature method (Quadra) gains speed advantage as the number of gaussians in the mixture increases because the computational cost of approximation using quadrature does not increase with the number of Gaussian components in the mixture. The speed advantage of the quadrature method becomes more salient as we have more Gaussians in the mixture which is reflected in the growing difference among the running times of these methods. Since the number of gaussians in the mixture (N) is determined by the number of hyperparameter samples we use for marginalisation in our algorithm, if we want to use a larger number of hyperparameter sets, we should adopt the quadrature method for fast approximation of the Gaussian mixture entropy at decent accuracy.

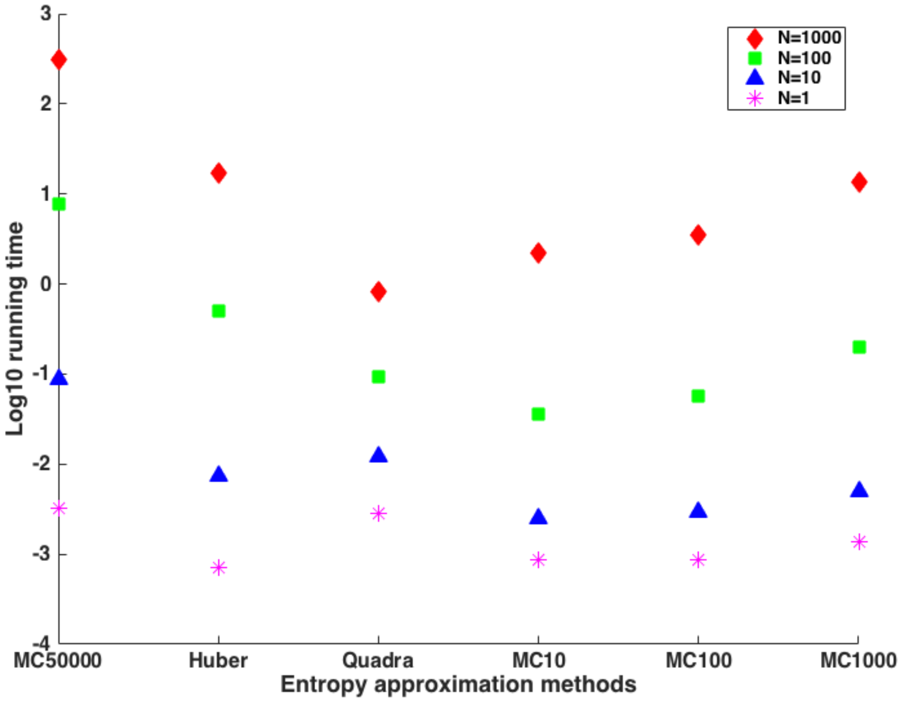

Figure 3.5: *Compare the running times of the approximation methods.*

4 Experiments and Results

In this chapter, we will implement ESBOPA approach described above and compare its performance with some of the existing acquisition functions introduced in Section 2.1.2. The first set of experiments are within model whereby ESBOPA methods will be tested using 1D and 2D test functions generated from a predetermined Gaussian process model. The second set of experiments will then evaluate ESBOPA algorithm with well-known benchmark test functions such as Branin-Hoo (2D), Rosenbrock(2D) as well as Hartmann-6 (6D).

In all experiments, we use Metropolis Hastings algorithm [5] to sample hyperparameters $\boldsymbol{\theta}$ and η and then marginalise all acquisition functions over the hyperparameters and η. The samples of hyperparameters and η must fulfilll two conditions: 1) hyperparameters such as the characteristic length scale, output scale need to be positive and 2) η is not greater than the minimum observation y_{min}. In order to enforce these two conditions, we assume the prior distributions of $\log \boldsymbol{\theta}$ and $\eta - y_{min}$ are broad Gaussian distributions. We then generate M samples of $\{\log \boldsymbol{\theta}^{(j)}, \eta^{(j)} - y_{min} | j = 1, \dots, M\}$, from which the samples of $\{\boldsymbol{\theta}^{(j)}, \eta^{(j)} | j = 1, \dots, M\}$ are reconstructed.

The posterior distribution $p(\boldsymbol{\Psi}|D_n)$, where $\boldsymbol{\Psi} = \{\boldsymbol{\theta}, \eta\}$, is an essential element for the sampling process. We assume hyperparameters and η are all independent and obtain the following expression by Bayes rule:

$$p(\boldsymbol{\Psi}|D_n) \propto p(D|\boldsymbol{\Psi}) p(\boldsymbol{\theta}) p(\eta)$$

where

$$p(\boldsymbol{\theta}) = \left| \frac{\mathrm{d} \log \boldsymbol{\theta}}{\mathrm{d}\boldsymbol{\theta}} \right| p(\log \boldsymbol{\theta}) = \frac{1}{\boldsymbol{\theta}} p(\log \boldsymbol{\theta}),$$

$$p(\eta) = p(y_{min} - \eta) = \left| \frac{\mathrm{d} \log(y_{min} - \eta)}{\mathrm{d}\eta} \right| p\big(\log(y_{min} - \eta)\big) = \frac{1}{(y_{min} - \eta)} p\big(\log(y_{min} - \eta)\big).$$

Therefore, the logarithm of the unnormalised posterior distribution $\log \tilde{p}(\boldsymbol{\Psi}|D_n)$, from which we draw samples of hyperparameters and η, has the form:

$$\log \tilde{p}(\boldsymbol{\Psi}|D_n) = \log p(D_n|\boldsymbol{\Psi}) + \log p(\boldsymbol{\theta}) + \log p(\eta)$$

where

$$\log p(D_n|\boldsymbol{\Psi}) = -\frac{1}{2} \mathbf{y}_n^T [K_f(\mathbf{X}_n, \mathbf{X}_n') + \sigma_n^2 \mathbf{I}]^{-1} \mathbf{y}_n - \frac{1}{2} \log |K_f(\mathbf{X}_n, \mathbf{X}_n') + \sigma_n^2 \mathbf{I}| - \frac{n}{2} \log 2\pi,$$

$$\log p(\boldsymbol{\theta}) = \log \left[\frac{1}{\boldsymbol{\theta}} p(\log \boldsymbol{\theta}) \right] = \log p(\log \boldsymbol{\theta}) - \log \boldsymbol{\theta},$$

$$\log p(\eta) = \log \left[\frac{1}{(y_{min} - \eta)} p\big(\log(y_{min} - \eta)\big) \right] = \log p\big(\log(y_{min} - \eta)\big) - \log(y_{min} - \eta).$$

In addition to hyperparameter sampling, we uses two common metrics to evaluate the performance of various Bayesian optimisation algorithms in all experiments [12]. The first metric is Immediate regret (IR) which is defined as:

$$IR = |f(\mathbf{x}_*) - f(\hat{\mathbf{x}}_n)| \tag{4.1}$$

where $\mathbf{x}_*$ is the location of true global minimiser and $\hat{\mathbf{x}}_n$ is the location recommended by a Bayesian optimiser after n iterations, which corresponds to the global minimiser of the posterior mean[13]. The second metric is the Euclidean distance of an optimiser's best recommendation $\hat{\mathbf{x}}_n$ from true global minimiser $\mathbf{x}_*$, which is defined as

$$\|L\|_2 = \|\mathbf{x}_* - \hat{\mathbf{x}}_n\|. \tag{4.2}$$

4.1 Within-Model Comparison

We first carried out within-model experiments over 1D and 2D unit domains (i.e $\mathcal{X} = [0, 1]$ and $\mathcal{X} = [0, 1]^2$). The methodology of these experiments is similar to those adopted in [13] [12] and is summarised below:

[1] Each test function is generated by sampling m function values from the Gaussian process prior with a squared-exponential covariance function. The hyperparmeter values, namely length scale l, output variance γ^2 and noise variance σ_n^2 are specified by us.

[2] M test functions are generated from the resulting Gaussian process posterior mean.

[3] Evaluate the test function using three optimisation algorithms (PI,EI, ESBOPA) by MC sampling hyperparameter values. All the methods are initialized with three random measurements collected using latin hypercube sampling [1].

[4] After each iteration, each algorithm returns the global minimiser of the posterior mean over the test function $\hat{\mathbf{x}}_n$ as the best recommendation of the true minimiser $\mathbf{x}_*$. Evaluate the best recommendation for the minimum of the test function $f(\hat{\mathbf{x}}_n)$.

[5] Compute the immediate regret $IR = |f(\mathbf{x}_*) - f(\hat{\mathbf{x}}_n)|$ and the Euclidean distance $\|L\|_2 = \|\mathbf{x}_* - \hat{\mathbf{x}}_n\|$. Repeat the above procedures for all M test functions and plot the median of the IR against the number of iterations. Confidence bands equal to one standard deviation are obtained using the bootstrap method.

Lobato et al. [13] carried out within model experiments by fixing hyperparamters to the predetermined values but in our within model experiments, we treat hyperparameters as unknown, which is more realistic, and marginalise over them during optimisation.

4.1.1 1D Test Functions

This experiment is conducted over 50 different 1D test functions which are randomly generated using the procedures described above. Specifically, 50 samples of function values are used (m=50) and the hyperparmeters adopted for the GP prior are $l_1^2 = 0.01$, $\gamma^2 = 1$, $\sigma_n^2 = 10^{-6}$. Figure **??** shows six examples of the synthetic test functions. It is evident that almost all test functions generated via the within model approach are multimodal. Three optimisation algorithms are used, namely ESBOPA, EI and PI and the optimisation of each test function starts from 1 initial observation data.

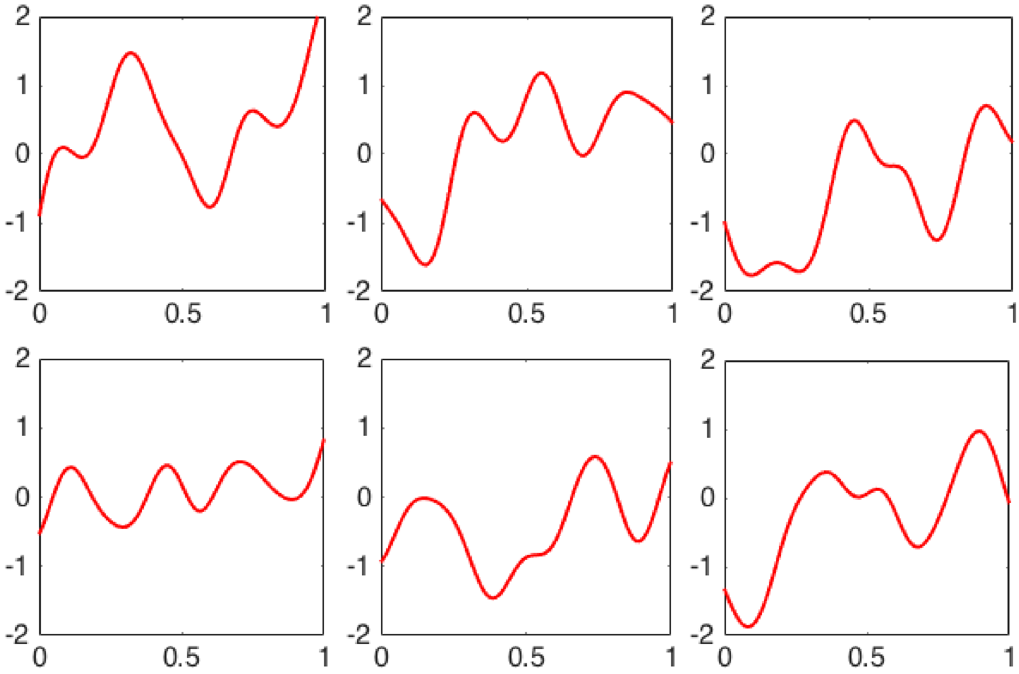

Figure 4.1: *Examples of 1D within model test functions. All test function is constructed using a GP prior with a squared-exponential covariance function. The hyperparmeters of the GP model are* $l_1^2 = 0.01$, $\gamma^2 = 1$, $\sigma_n^2 = 10^{-6}$

Figure **??** and Figure **??** illustrate the resultant immediate regret and the Euclidean distance for all three algorithms starting from the first evaluation. For both evaluation metrics, ESBOPA beats PI throughout all 50 consecutive evaluations. ESBOPA initially loses out to EI but is able to catch up after 20 iterations. This may be due to PI is more exploitation-oriented than EI and thus tends to be trapped at local minimum during the initial iterations. All three algorithms converge to a similar constant level of accuracy eventually because the test functions are relatively simple.

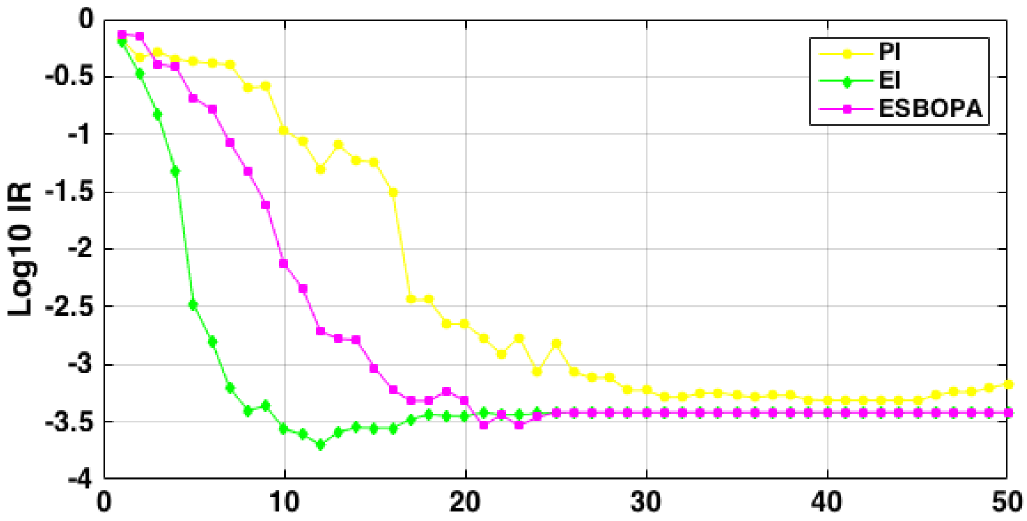

Figure 4.2: *Median immediate regrets of ESBOPA, EI and PI in the 1D within model experiments over 50 test functions.*

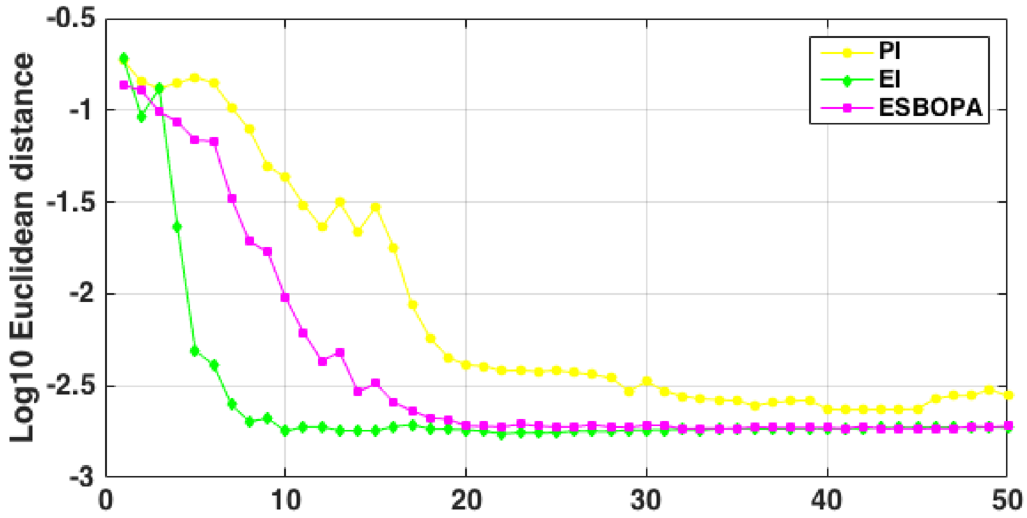

Figure 4.3: *Median Euclidean distance between the global minimiser* $\mathbf{x}_*$ *and the best recommendation* $\hat{\mathbf{x}}_n$ *of ESBOPA, EI and PI in the 1D within model experiments over 50 test functions*

4.1.2 2D Test Functions

Similar to the 1D case, this experiment is carried out over 60 within model 2D test functions. These test functions are generated by sampling 50 function values (m=50)and setting the hyperparmeters of the GP prior to $l_1^2 = l_2^2 = 0.1$, $\gamma^2 = 1$, $\sigma_n^2 = 10^{-6}$. Figure **??** shows three examples of such 2D test functions. The optimisation processes for all algorithms start from 3 initial observation data.

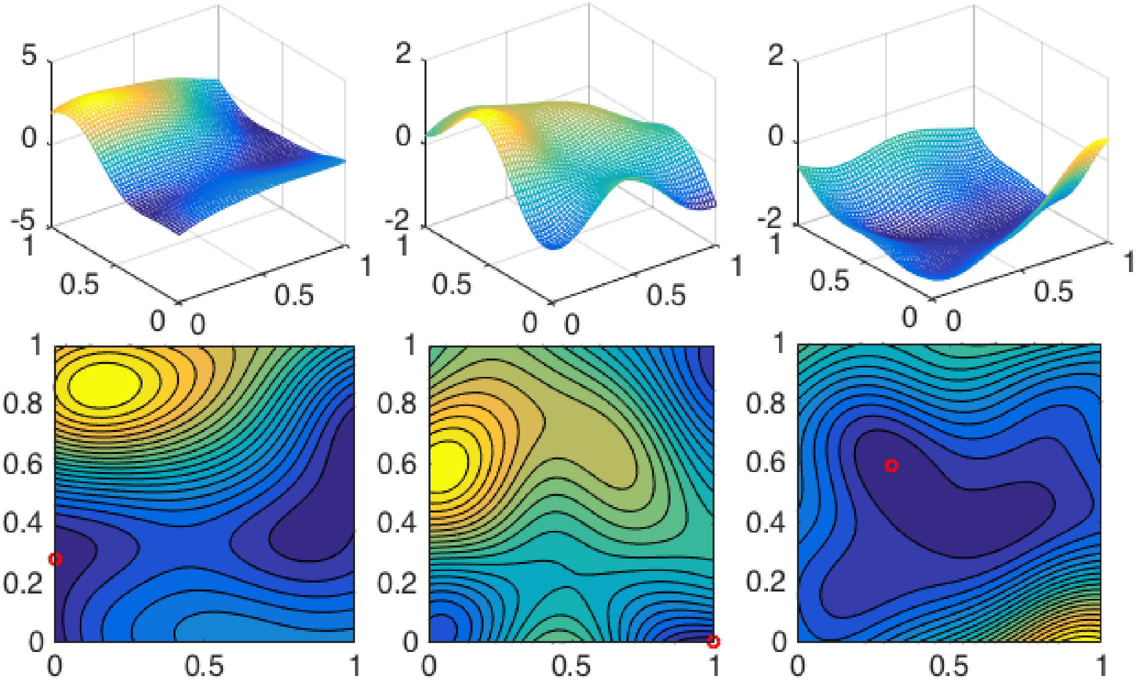

Figure 4.4: Examples of 2D within model test functions which are generated using a GP prior with a squared-exponential covariance function. The hyperparmeters of the GP model are $l_1^2 = l_2^2 = 0.1$, $\gamma^2 = 1$, $\sigma_n^2 = 10^{-6}$. Subfigures in the top row are the 3D mesh plots of the three test functions while their corresponding contour plots are shown in the bottom row.The location of the true global minimiser for each test function is indicated with a red circle.

The median performance for all three algorithms over all test functions are displayed in Figure **??** and Figure **??**. From the graphs, the ESBOPA method beats PI consistently but only gains an advantage over EI after taking enough number of iterations (over 40 iterations). One possible explanation for this initial poor performance of ESBOPA is that EI has comparatively more exploitative searching behaviours while ESBOPA requires more evaluations to gain information about the minimiser at the initial stage. More importantly, despite the quick convergence rate of EI, its searching accuracy stops at order of $IR \approx 10^{-3}$ and $\|L\|_2 \approx 10^{-2}$ after around 20 iterations. On the other hand, although ESBOPA converges at a slower rate, its accuracy keeps decreasing even after 45 iterations which is a promising sign that ESBOPA method could give more accurate results at the expense of convergence speed.

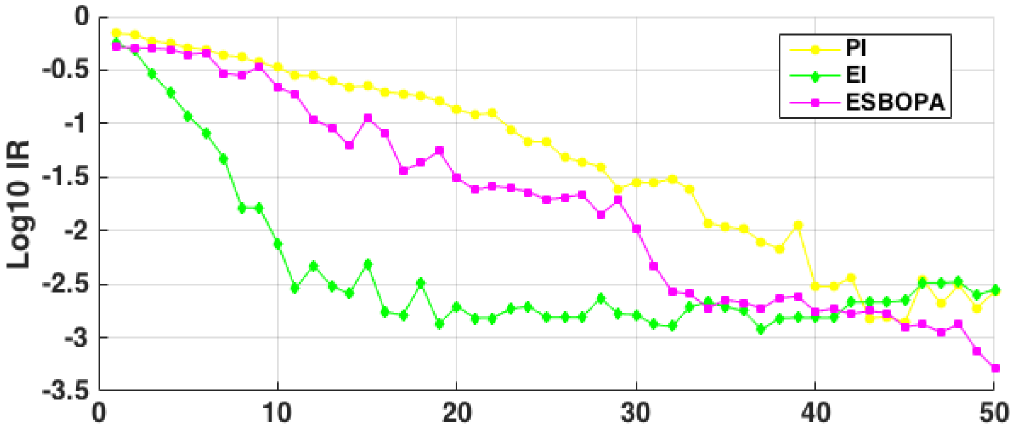

Figure 4.5: *Median immediate regrets of ESBOPA, EI and PI in the within model experiments over 2D domain.*

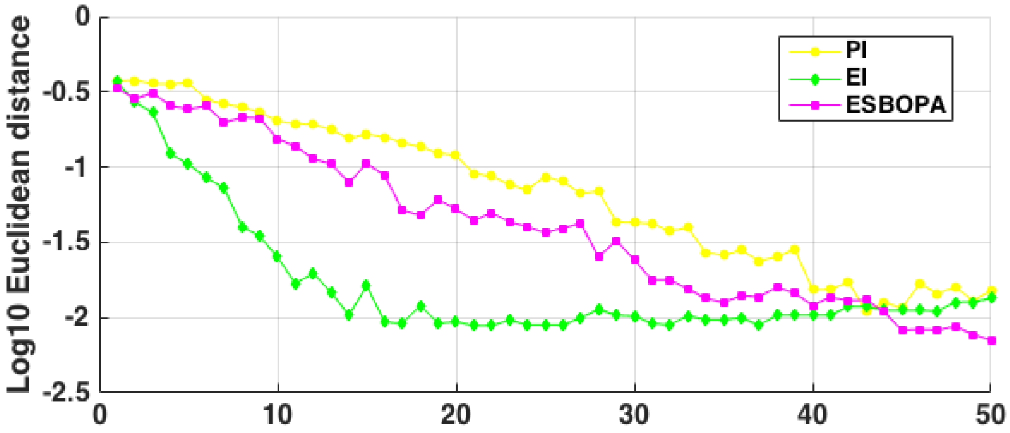

Figure 4.6: *Median Euclidean distance between the global minimiser* $\mathbf{x}_*$ *and the best recommendation* $\hat{\mathbf{x}}_n$ *of ESBOPA, EI and PI in the within model experiments over 2D domain.*

4.2 Benchmark Test Functions

In this section, we conduct another set of experiments using more challenging test functions such as Branin 2D, Rosenbrock 2D and Hartmann 6D. In all tests, the observation noise are set to a relatively low level $\sigma_n^2 = 10^{-3}$.

4.2.1 Branin 2D

We use a modified version of the Branin 2D function which has the form:

$$f(\mathbf{x}) = \frac{1}{10}\left[\left(\tilde{x}_2 - \frac{5.1}{4\pi^2}\tilde{x}_1^2 + \frac{5}{\pi}\tilde{x}_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(\tilde{x}_1) + 10\right] - 15 \tag{4.3}$$

where $\tilde{x}_1 = 15x_1 - 5$ and $\tilde{x}_2 = 15x_2$.

The modified Branin function has exactly the same shape and minimiser locations as the original Branin function as shown in Figure ??. However, its output range changes from [0.3979, 308.1291] to [-14.9602, 15.8129] and its input scale changes from $x_1 \in [-5, 10], x_2 \in [0, 15]$ to $x_1, x_2 \in [0, 1]$. The global minimum of the modified Branin 2D function is -14.9602 at $\mathbf{x}_* = [0.1239, 0.8183], [0.5428, 0.1517]$ and $[0.9617, 0.1650]$.

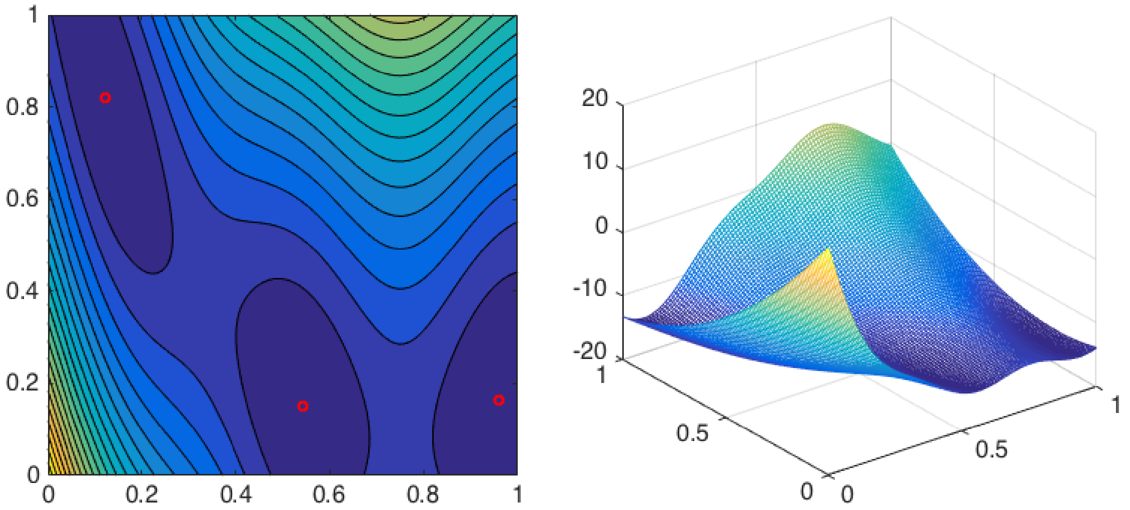

Figure 4.7: *The modified Branin 2D function with the globally minimum value of -0.9973 at three locations* $\mathbf{x}_* = [0.1239, 0.8183], [0.5428, 0.1517]$ *and* $[0.9617, 0.1650]$*, denoted by red circles.*

The experiments with the Branin 2D function start from three initial observations. The performance of ESBOPA approach is compared with those of PI and EI in Figure ?? and Figure ??. Similar to the case for the 2D within model experiments, ESBOPA experiences a slow convergence rate, thus losing out to EI and PI at the initial stage. However, as more evaluations are taken, the performance of both improvement-based methods start to stabilise while the optimisation error of ESBOPA continues to decrease. This enables ESBOPA to catch up EI and PI and surpass them slightly in the end.

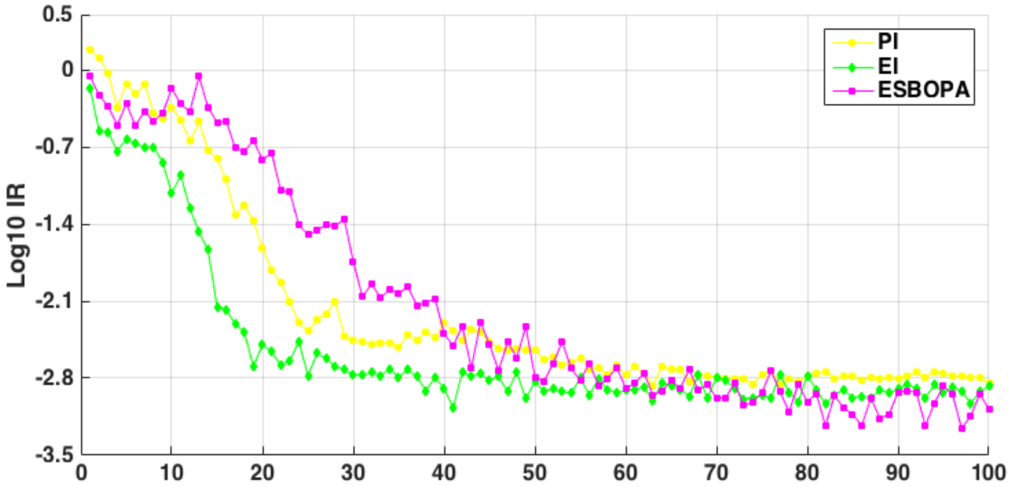

Figure 4.8: *Median immediate regrets of ESBOPA, EI and PI in the Branin 2D problem.*

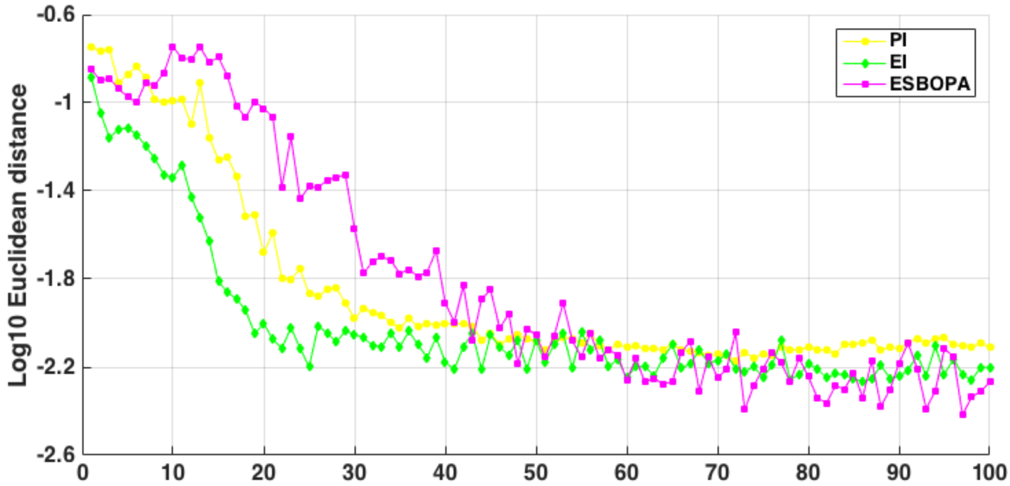

Figure 4.9: *Median Euclidean distance between the global minimiser* $\mathbf{x}_*$ *and the best recommendation* $\hat{\mathbf{x}}_n$ *of ESBOPA, EI and PI in the Branin 2D problem.*

One interesting point we would like to illustrate through the Branin problem is the fundamentally different mechanisms behind ESBOPA and improvement-based approaches. As shown in Figure **??**, the locations for evaluation proposed by ESBOPA spread across the whole domain but those by EI and PI tend to quickly concentrate at the zones of low functional values. Yet, ESBOPA is still able to predict the location of the global minimimum from such scattered evaluation samples. This different querying behaviour is due to the fact that information-based approaches such as ESBOPA aim to select the query point that maximises the information gain about the minimiser but improvement-based approaches such as EI and PI choose the query point that leads to an improvement over the current best function value observed [2].

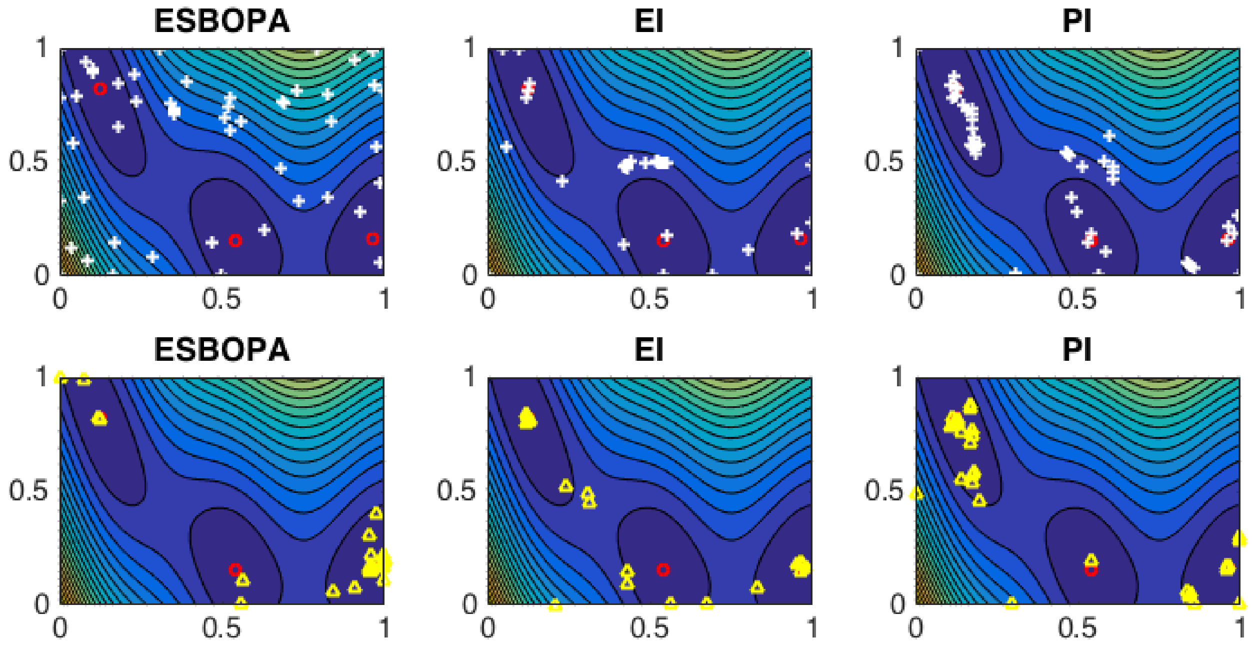

__Figure 4.10:__ 50 evaluations taken by ESBOPA, EI and PI in the Branin 2D problem. In the top row row of subfigures, the white crosses indicate the consecutive 50 evaluation samples proposed by the three algorithms. In the bottom row of subfigures, the yellow triangles indicate the 50 best guesses of the global minimiser recommended by all three algorithms after each corresponding evaluation.

4.2.2 Rosenbrock 2D

The Rosenbrock 2D function, also known as Rosenbrock's banana function, contains its global minimum inside a long narrow valley, which makes optimisation difficult. In our experiments, we modified the function to have an input scale of $[-1,1]^2$ and an output range of $[-10,9]$. The modified Rosenbrock 2D function preserves the valley shape as shown in Figure **??** and has the form:

$$f(\mathbf{x}) = a(1-\tilde{x}_1)^2 + b(\tilde{x}_2 - \tilde{x}_1^2)^2 - c \tag{4.4}$$

where $\tilde{x}_1 = 2x_1$, $\tilde{x}_2 = 2x_2$, $a = {}^1\!/_{200}$, $b = {}^1\!/_2$, $c = 10$. The global minimum of the modified Rosenbrock 2D function is -10 at $\mathbf{x}_* = [0.5, 0.5]$.

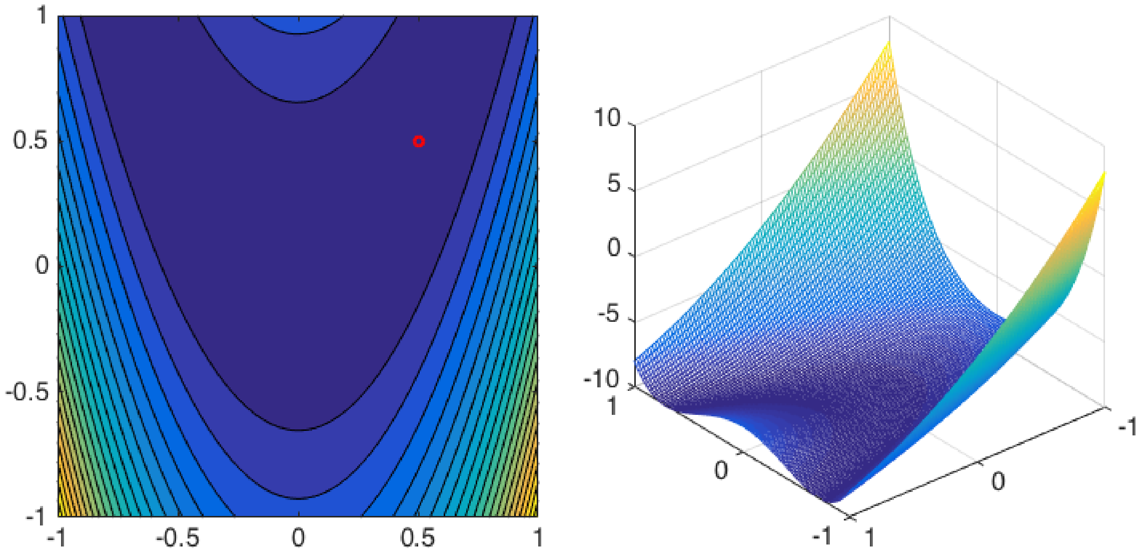

Figure 4.11: *The modified Rosenbrock 2D function with the globally minimum value of -10 at* $\mathbf{x}_* = [0.5, 0.5]$*, denoted by the red circle.*

As shown in Figure **??**, ESBOPA outperforms EI and PI after 30 evaluations and then stays better than the other two approaches in searching the minimum off the Rosenbrock function. The advantage of ESBOPA over EI and PI is more obvious in predicting the location of the global minimiser $\mathbf{x}_*$ as shown in Figure **??**. In this case, although the greedy improvement-based approaches can quickly identify the parabolic flat valley in the problem, entropy-based approach does a better job in searching the global minimum in the valley where function values are very close.

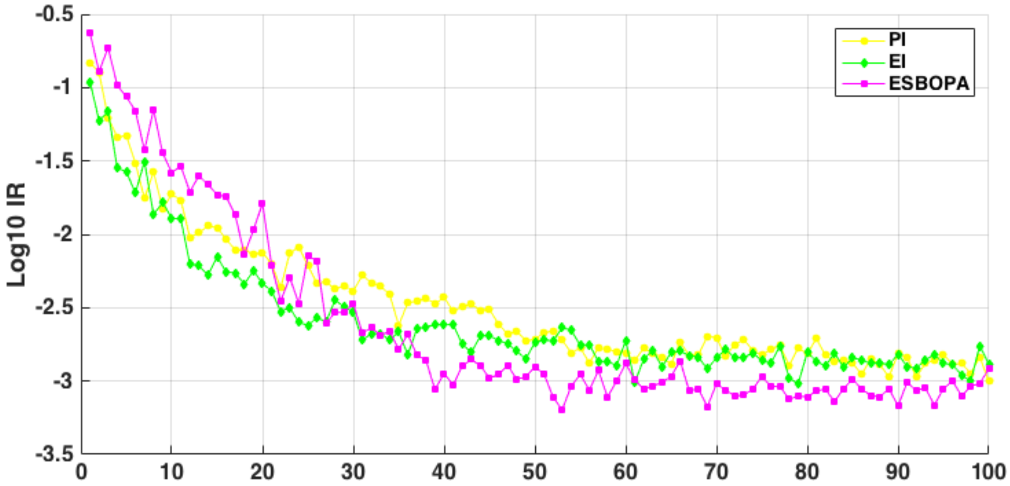

Figure 4.12: *Median immediate regrets of ESBOPA, EI and PI in the Rosenbrock 2D problem.*

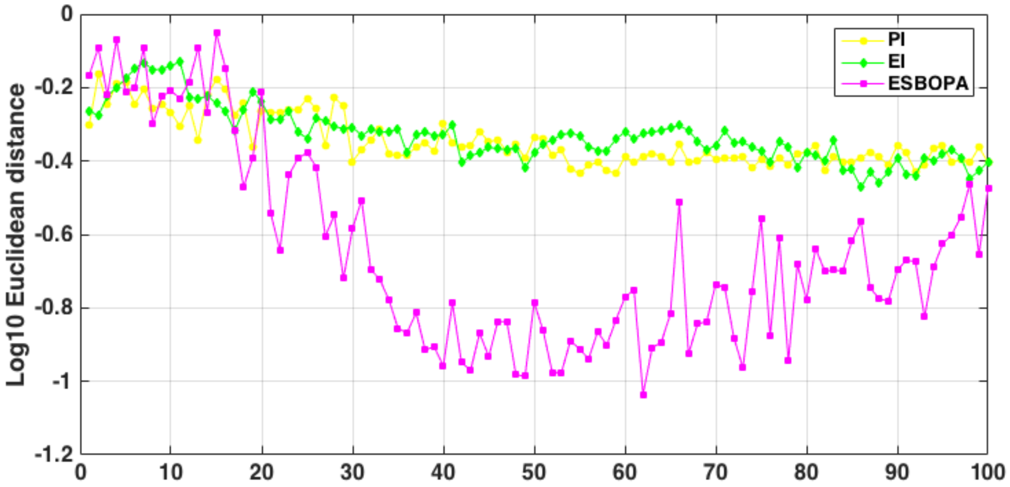

Figure 4.13: *Median Euclidean distance between the global minimiser* $\mathbf{x}_*$ *and the best recommendation* $\hat{\mathbf{x}}_n$ *of ESBOPA, EI and PI in the Rosenbrock 2D problem.*

4.2.3 Hartmann 6D

To further generalise our tests, we apply these algorithms to an even higher dimensional problem - Hartmann 6-D function (defined in $[0,1]^6$). The modified formula of Hartmann 6D has the following form:

$$f(\mathbf{x}) = 1.5 - \sum_{i=1}^{4} \alpha_i \exp\left[-\sum_{j=1}^{6} A_{ij}(x_j - P_{ij})^2\right], \quad \text{with} \tag{4.5}$$

$$\alpha = [1.0, 1.2, 3.0, 3.2]^T$$

$$\mathbf{A} = \begin{bmatrix} 10 & 3 & 17 & 3.5 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 1.7 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 14 \end{bmatrix}$$

$$\mathbf{P} = \begin{bmatrix} 0.1312 & 0.1696 & 0.5569 & 0.0124 & 0.8283 & 0.5886 \\ 0.2329 & 0.4135 & 0.8307 & 0.3736 & 0.1004 & 0.9991 \\ 0.2348 & 0.1451 & 0.3522 & 0.2883 & 0.3047 & 0.6650 \\ 0.4047 & 0.8828 & 0.8732 & 0.5743 & 0.1091 & 0.0381 \end{bmatrix}.$$

where $x_i \in [0,1]$ for all $i = 1, \ldots, 6$. Thus, the Hartmann 6-D function has a global minimum of -1.8224 at $\mathbf{x}_* = [0.20169, 0.150011, 0.476874, 0.275332, 0.311652, 0.6573]$

The tests start from 9 initial observations because of the higher dimension of the Hartmann function used. The optimisation performances of ESBOPA, EI and PI are presented in Figure **??** and **??**. In terms of immediate regrets, PI demonstrates the best performance followed by EI. ESBOPA has an initial advantage over EI but is overtaken by EI after around 60 iteration. Similar results are observed in [13] as PES delivers consistently worse performance than EI in the Hartmann 6D functions.

One explanation for the relatively poor performance of ESBOPA and PES in this problem is that information-theoretic approaches take more iterations to explore in high dimensional space, enabling the more greedy improvement-based approaches to gain advantage in optimising the relatively simple Hartmann function [13].

However, in the input space (Figure **??**), the best guess by ESBOPA is consistently closer to the true minimiser in terms of Euclidean distance than that predicted by EI. ESBOPA also perform better than PI before 60 evaluations in searching the minimiser location.

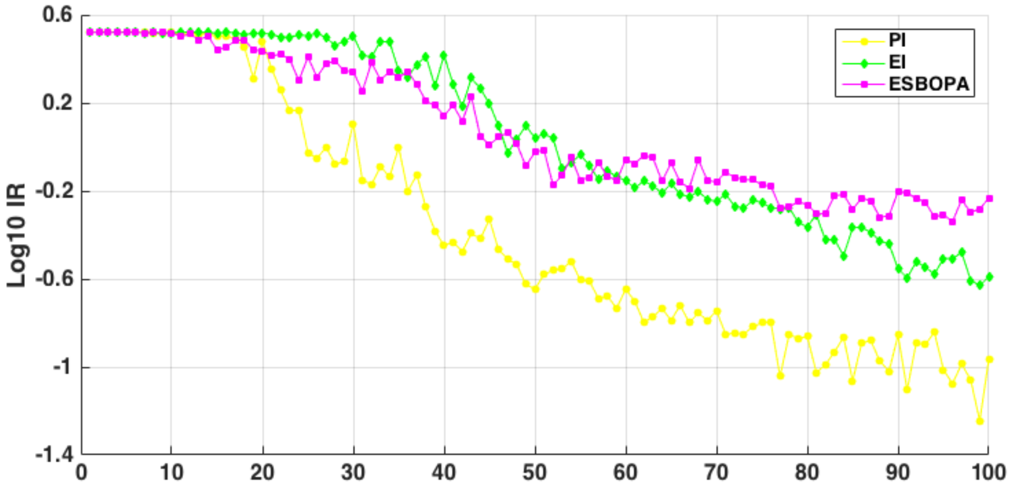

Figure 4.14: *Median immediate regrets for ESBOPA, EI and PI in the experiments with Hartmann 6D.*

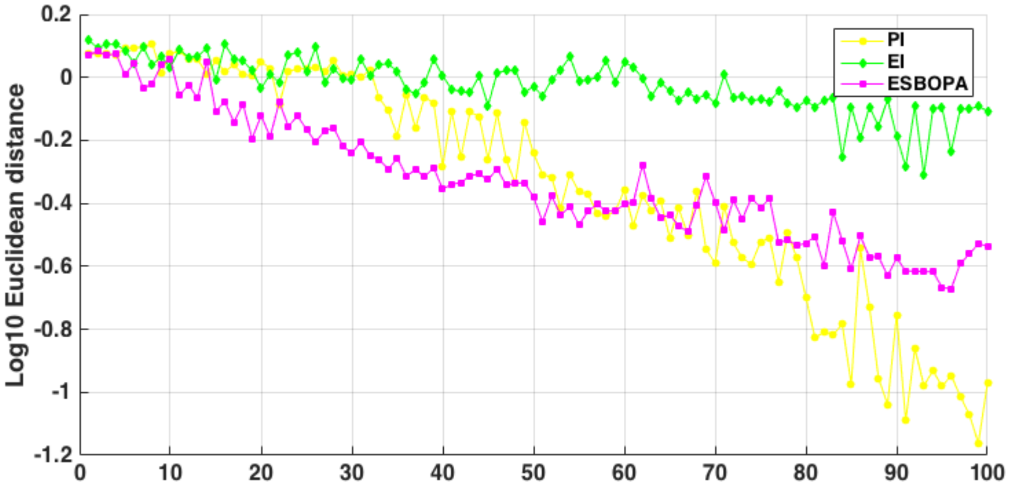

Figure 4.15: *Median Euclidean distance between the global minimiser $\mathbf{x}_*$ and the best recommendation $\hat{\mathbf{x}}_n$ for ESBOPA, EI and PI in the experiments with Hartmann 6D.*

4.3 The Effect of Noise on Performance

In the above experiments, the measurement noise level is set to a negligibly low level of $\sigma^2 = 10^{-3}$ or 10^{-6}. However, in real world, the observation noise might be non-trivial and a good optimisation algorithm should be able to find the global minimiser even in the presence of relatively high noise level. Therefore, in this section, we would test and compare the optimisation performance of different acquisition functions under the effect of different noise levels.

Figrue **??** shows that the optimisation results of ESBOPA, EI and PI acquisition functions applied to the Rosenbrock 2D problem at different noise levels. As the variance of observation noise σ_n^2 increases from 0.001 to 0.1, ESBOPA is still able to accurately identify the unknown global minimiser $\mathbf{x}_*$ and its performance stays close to those of EI and PI. However, EI and PI seem to do slightly better job than ESBOPA under the effect of high noise level $\sigma_n^2 = 10^{-1}$. This advantage of EI and PI under high observation noise is more evident in the case of the Branin 2D problem (Figure **??**).

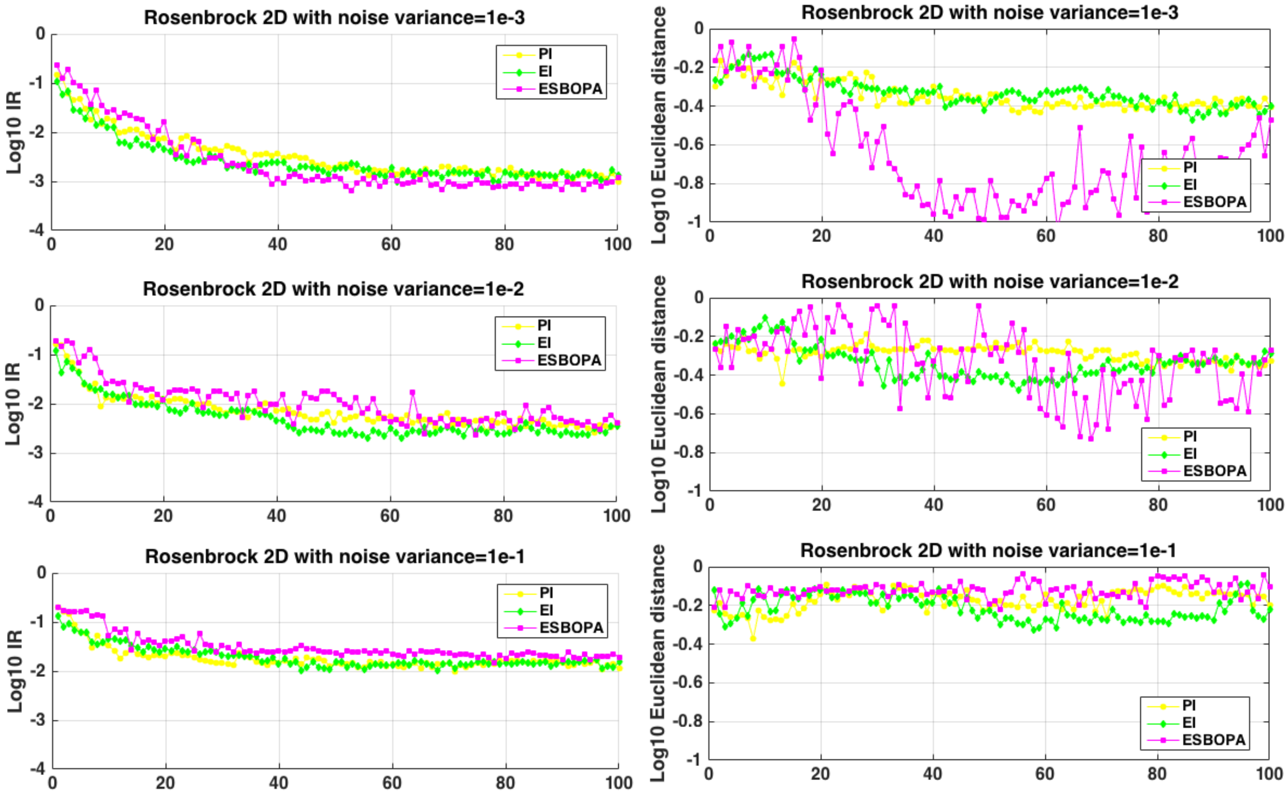

__Figure 4.16:__ Comparison of ESBOPA, EI and PI in the Rosenbrock 2D problem with noise levels 0.001, 0.01, 0.1. Subfigures on the left column show the results of the median immediate regret. Subfigures on the right column show the results of the median Euclidean distance between the global minimiser $\mathbf{x}_*$ and the best recommendation $\hat{\mathbf{x}}_n$ regret.

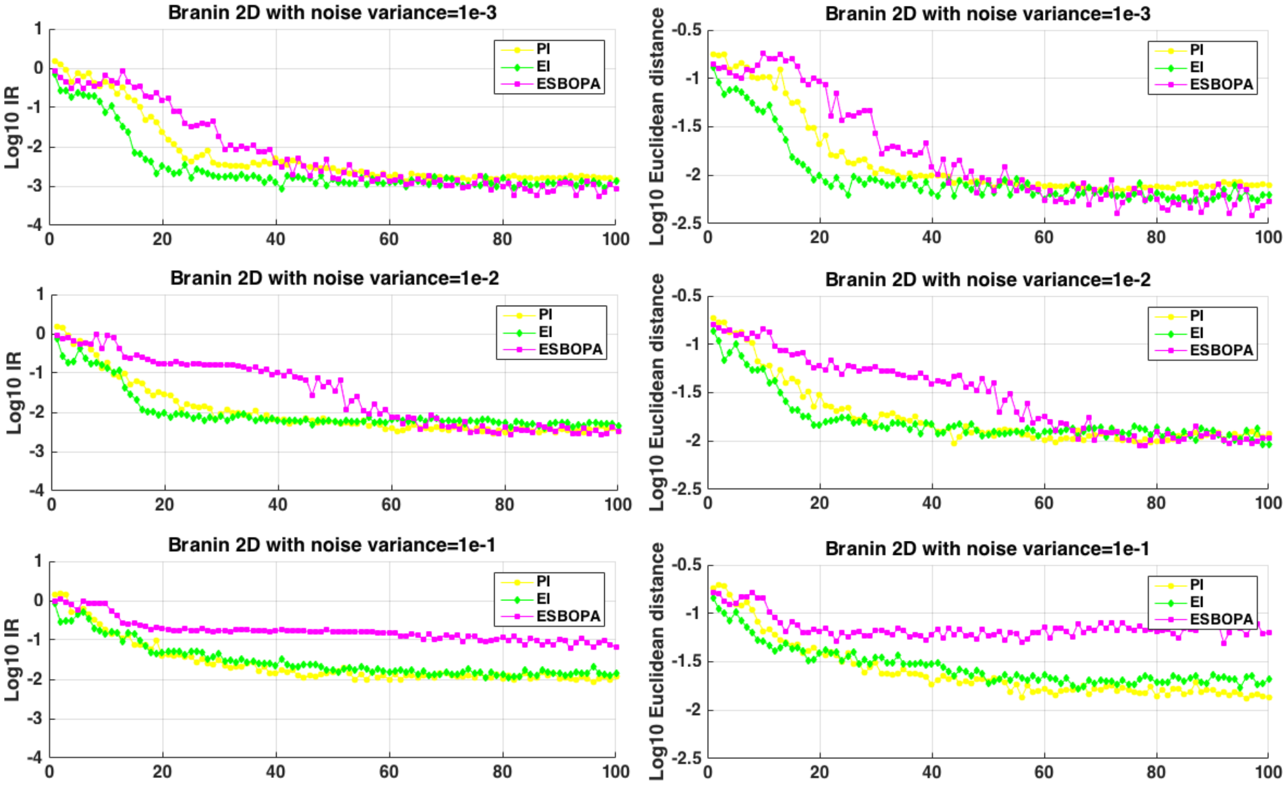

Figure 4.17: *Comparison of ESBOPA, EI and PI in the Branin 2D problem with noise levels 0.001, 0.01, 0.1. Subfigures on the left column show the results of the median immediate regret. Subfigures on the right column show the results of the median Euclidean distance between the global minimiser* $\mathbf{x}_*$ *and the best recommendation* $\hat{\mathbf{x}}_n$ *regret.*

5 Conclusion

The project successfully develops a novel entropy based algorithm, called ESBOPA in short, for Bayesian optimisation. With the creative use of the parabolic approximation and the hyperparameter η, ESBOPA approach requires less sampling efforts and fewer approximations in its implementation as compared to existing information-theoretic approaches such as Entropy Search (ES) and Predictive Entropy Search (PES). Meanwhile, ESBOPA approach is also compatible with a wider range of covariance functions and enjoys the merit of working in the one-dimensional output space. Our ESBOPA method is tested via within model experiments and several challenging benchmark optimisation problems. On the whole, the results demonstrate a desirable and very promising optimisation performance in comparison with the two popular acquisition functions, Expected Improvement and Probability of Improvement. Therefore, ESBOPA method offers a very competitive alternative to the existing heuristics for Bayesian optimisation and our project makes a concrete contribution to the development of the information-based approaches.

Our research work can definitely be extended in the many areas. First, if given more time, we would test ESBOPA with more complex real world optimisation problems and compare ESBOPA with a wider range of current heuristics such as PES, ES and random search. Second, ESBOPA approach relies heavily on the prior distribution over η which explicitly represent the global minimum. In our project, we just use a gaussian distribution for η and set the hyperparameters defining the distribution (i.e. mean and variance) arbitrarily. However, another optimisation process can be applied to fine-tune the hyperparameters defining the η distribution and other appropriate distributions such as gamma distribution are also worth exploring. With a more well-defined η distribution, the performance of ESBOPA can be further enhanced. In addition, ESBOPA approach can also be improved by incorporating derivative observations at η.

Bibliography

[1] Eric Brochu, Vlad M Cora, and Nando De Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.

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

[3] Jonas Mockus. Application of bayesian approach to numerical methods of global and stochastic optimization. *Journal of Global Optimization*, 4(4):347–365, 1994.

[4] Carl Edward Rasmussen. Gaussian processes for machine learning. 2006.

[5] Christopher M Bishop. Pattern recognition. *Machine Learning*, 128, 2006.

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

[7] Donald R Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of global optimization*, 21(4):345–383, 2001.

[8] Laurence Charles Ward Dixon and Giorgio P Szegö. *Towards global optimisation.* North-Holland Amsterdam, 1978.

[9] Daniel James Lizotte. *Practical bayesian optimization.* University of Alberta, 2008.

[10] Dennis D Cox and Susan John. A statistical method for global optimization. In *Systems, Man and Cybernetics, 1992., IEEE International Conference on*, pages 1241–1246. IEEE, 1992.

[11] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

[12] Philipp Hennig and Christian J Schuler. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13(Jun):1809–1837, 2012.

[13] José Miguel Hernández-Lobato, Matthew W Hoffman, and Zoubin Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. In *Advances in neural information processing systems*, pages 918–926, 2014.

[14] Salomon Bochner. *Lectures on Fourier Integrals.(AM-42)*, volume 42. Princeton University Press, 2016.

[15] Ali Rahimi, Benjamin Recht, et al. Random features for large-scale kernel machines. In *NIPS*, volume 3, page 5, 2007.

[16] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

[17] Jarno Vanhatalo, Jaakko Riihimäki, Jouni Hartikainen, and Aki Vehtari. Bayesian modeling with gaussian processes using the matlab toolbox gpstuff. *submitted. http://becs. aalto. fi/en/research/bayes/gpstuff/GPstuffDoc31. pdf*, 2011.

[18] Zi Wang and Stefanie Jegelka. Max-value entropy search for efficient bayesian optimization. *arXiv preprint arXiv:1703.01968*, 2017.

[19] Tom Gunter, Michael A Osborne, Roman Garnett, Philipp Hennig, and Stephen J Roberts. Sampling for inference in probabilistic models with fast bayesian quadrature. In *Advances in neural information processing systems*, pages 2789–2797, 2014.

[20] Marco F Huber, Tim Bailey, Hugh Durrant-Whyte, and Uwe D Hanebeck. On entropy approximation for gaussian mixture random vectors. In *Multisensor Fusion and Integration for Intelligent Systems, 2008. MFI 2008. IEEE International Conference on*, pages 181–188. IEEE, 2008.