

Randomised Gaussian Process Upper Confidence Bound for Bayesian Optimisation

Julian Berk*, Sunil Gupta, Santu Rana and Svetha Venkatesh

Applied Artificial Intelligence Institute

{jumberk, sunil.gupta, santu.rana, svetha.venkatesh}@deakin.edu.au

Abstract

In order to improve the performance of Bayesian optimisation, we develop a modified Gaussian process upper confidence bound (GP-UCB) acquisition function. This is done by sampling the exploration-exploitation trade-off parameter from a distribution. We prove that this allows the expected trade-off parameter to be altered to better suit the problem without compromising a bound on the function’s Bayesian regret. We also provide results showing that our method achieves better performance than GP-UCB in a range of real-world and synthetic problems.

1 Introduction

Global optimisation is a cornerstone of modern scientific innovation. Optimisation of alloys and other materials has allowed us to create massive vehicles that are both strong and light enough to fly. Optimisation in medical science has seen us live longer and healthier lives than previously thought possible. This optimisation usually involves a trial-and-error approach of repeated experiments with different inputs to determine which input produces the most desirable output. Unfortunately, many systems are expensive to sample, and the heuristic methods commonly used to select inputs are not sample-efficient. This can lead to these optimisation experiments being prohibitively costly. As such, methods that can select inputs in a sample-efficient manner can lead to faster and cheaper innovation in a wide range of fields.

Bayesian optimisation is one of the most sample efficient methods for optimising expensive, noisy systems [Brochu *et al.*, 2010; Shahriari *et al.*, 2015]. It has shown excellent performance on a range of practical problems, including problems in biomedical science [Turgeon *et al.*, 2016; Gonzalez *et al.*, 2015], materials science [Li *et al.*, 2017; Kikuchi *et al.*, 2018; Ju *et al.*, 2017], and machine learning [Snoek *et al.*, 2012; Klein *et al.*, 2017; Xia *et al.*, 2017]. It does so by using the data from previous samples to generate a statistical model of the system. This model is then used to suggest the next input through an *acquisition function*. The

design of the acquisition function is non-trivial; and is critically important to the algorithm’s performance. It must balance selecting points with the goal of improving the statistical model (*exploration*), and selecting points with the goal of utilising the improved statistical model to find the global optima (*exploitation*). In addition to this, the costly nature of these problems means that it is desirable to have a theoretical guarantee of the algorithm’s performance.

There are a large range of acquisition functions, all with different balances of exploration and exploitation, but we focus on *Gaussian process upper confidence bound* (GP-UCB) [Srinivas *et al.*, 2010] in this work. This controls its exploration-exploitation trade-off with a single hyperparameter, β_t . It has strong theoretical guarantees on the overall convergence rate, but the bound they give is fairly loose. This causes the value of β_t to be too large, causing significant over-exploration and hence poor practical performance. In practice, the theoretical guarantees need to be weakened by selecting a far smaller β_t .

We introduce a novel modification to the GP-UCB acquisition function that significantly improves its exploration-exploitation balance while still having a strong convergence guarantee. This is done by sampling its trade-off parameter with a distribution that allows for a range of exploration factors. However, the distribution is chosen such that convergence is guaranteed to be sub-linear while the sampled β_t is generally smaller than the traditional GP-UCB. This reduction leads to a direct improvement on the convergence efficiency in practice. We demonstrate this improved performance over the standard GP-UCB implementation in a range of both synthetic benchmark functions and real-world applications. In summary, our contributions are:

- The development of a modified acquisition function: RGP-UCB.
- A convergence analysis of Bayesian optimisation using RGP-UCB.
- The demonstration of the performance of our method on a range of synthetic and real-world problems.

2 Background

In this section, we provide a brief overview of Bayesian optimisation, with an emphasis on acquisition functions and regret bounds. For a more in-depth overview of Bayesian

*Contact Author

optimisation, we refer readers to [Rasmussen and Williams, 2006], [Shahriari *et al.*, 2015], and [Brochu *et al.*, 2010].

2.1 Bayesian Optimisation

Bayesian optimisation is a method for optimising expensive, noisy black-box functions. It represents the system being optimised as an unknown function, f . As this is black-box, it is impossible to directly observe. However, we can sample it with an input variable, x , to obtain a noisy output, $y = f(x) + \epsilon_{noise}$, where $\epsilon_{noise} \sim N(0, \sigma_{noise})$ is the random noise corrupting the measurement. Bayesian optimisation seeks to efficiently find the optimal input for such systems over a bounded search space, \mathcal{X} :

$$x^* = \arg \max_{x \in \mathcal{X}} f(x) \quad (1)$$

To do this, it creates a statistical model of f using all previously sampled input-output pairs, $D_{t-1} = \{x_i, y_i\}_{i=1}^{t-1}$. This statistical model is usually a *Gaussian process*, but other models can be used. The statistical model is then used to create an *acquisition function*, $\alpha(x)$. This is essentially a map of our belief of how useful a given input will be for optimising the system. As such, it can be used to suggest x_t ; the next input with which to sample the system. This input and its corresponding output can then be added to the previous data, $D_t = D_{t-1} \cup \{x_t, y_t\}$. This process can be iterated, with the Gaussian process improving with each iteration, until a stopping condition has been met. As Bayesian optimisation is generally used on expensive systems, this stopping condition is often a maximum number of experiments, T .

2.2 Gaussian Process

The Gaussian process is the most common statistical model used in Bayesian optimisation. It models each point in $x \in \mathcal{X}$ as a Gaussian random variable. As such, it is completely characterised by a mean function, $\mu(x)$, and a variance function, $\sigma^2(x)$. However, in order to model sensibly behaved functions there must be correlation between neighbouring points. This is done by conditioning the distribution on the data via a *kernel function*. There are many viable kernel functions, but one of the simplest and most popular is the *squared exponential kernel* as it only depends on a single hyperparameter, the lengthscale l . This is given by $k_{SE}(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2l^2}\right)$. Using a kernel such as this, a predictive distribution, $f(x) \sim \mathcal{N}(\mu(x), \sigma^2(x))$, can be obtained by conditioning the Gaussian process on the data, D_t :

$$\begin{aligned} \mu_t(x) &= \mathbf{k}_*(\mathbf{K}_t + \sigma_{noise}^2 \mathbf{I})^{-1} \mathbf{y}, \\ \sigma_t^2(x) &= k_{**} - \mathbf{k}_*(\mathbf{K}_t + \sigma_{noise}^2 \mathbf{I})^{-1} \mathbf{k}_*^T \end{aligned} \quad (2)$$

where $\mathbf{K}_{t,(i,j)} = k(x_i, x_j)$ is the *kernel matrix*, $k_{**} = k_t(x, x)$, and $\mathbf{k}_* = [k(x_1, x), k(x_2, x), \dots, k(x_t, x)]$. Here \mathbf{I} is the identity matrix with the same dimensions as \mathbf{K}_t , and σ_{noise} is the output noise standard deviation.

2.3 Acquisition Functions

Once the Gaussian process has been generated, it must then be used to create an *acquisition function*. This is chosen such

that its global maxima in \mathcal{X} will be the best next point to sample:

$$x_t = \arg \max_{x \in \mathcal{X}} \alpha_{t-1}(x) \quad (3)$$

However, the design of such a function is non-trivial. It must first suggest points spread over the search space to improve the Gaussian process. This is called *exploration*. Once the Gaussian process has been improved enough, it must then transition to suggesting points in regions that have a high probability of containing the global optima. This is called *exploitation*. If an acquisition function does not explore enough, it may get stuck exploiting sub-optimal regions and never find the global optima. However, if it explores too much, it may waste costly evaluations improving an already adequate Gaussian process. This makes balancing exploration and exploitation vital. There is a wide range of common acquisition functions, all of which have different balances of exploration and exploitation. These include entropy search (ES) by [Hennig and Schuler, 2012], predictive entropy search (PES) by [Hernández-Lobato *et al.*, 2014], knowledge gradient (KG) by [Scott *et al.*, 2011], and others. However, we will only consider the *Gaussian process upper confidence bound* (GP-UCB) by [Srinivas *et al.*, 2010], Thompson sampling by [Russo and Van Roy, 2014], max-value entropy search (MES) by [Wang and Jegelka, 2017], and *expected improvement* (EI) by [Jones *et al.*, 1998], with the latter three only being used as baselines.

GP-UCB

GP-UCB [Srinivas *et al.*, 2010] is one of the most intuitive acquisition functions. It balances exploration and exploitation through a single hyperparameter, β_t :

$$\alpha_t^{GP-UCB}(x) = \mu_t(x) + \sqrt{\beta_t} \sigma_t(x) \quad (4)$$

Increasing β_t makes the acquisition function favour points with high variance, causing more exploration. Decreasing β_t will make the acquisition function favour points with high mean, causing more exploitation. However, the selection of β_t is not done to optimally balance exploitation and exploration, but is done such that the cumulative regret is bounded. It has been proved that, assuming the chosen kernel satisfies

$$P \left\{ \sup_{x \in \mathcal{X}} |\partial f / \partial x_i| > L \right\} \leq a e^{-(L/b)^2}, \quad i = 1 \dots t \quad (5)$$

for some constants $a, b > 0$, then with probability $1 - \delta$, the algorithm will have sub-linear regret if

$$\beta_t = 2 \log(t^2 \pi^2 / (3\delta)) + 2d \log \left(t^2 d b r \sqrt{\log(4da/\delta)} \right) \quad (6)$$

While the regret bound provided by this choice of β_t is desirable, it unfortunately is far larger than needed. This leads to sub-optimal real world performance due to over-exploration. In their own paper, the authors divided the suggested β_t by a factor of 5 to achieve better performance [Srinivas *et al.*, 2010].

3 Proposed Method

We describe our improved GP-UCB acquisition function, *randomised Gaussian process upper confidence bound* (RGP-UCB), and prove that it has a sub-linear regret bound.

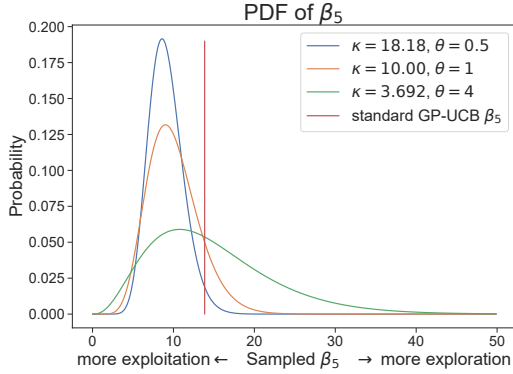


Figure 1: A range of the gamma distributions used to draw β_5 with $t = 5$. The parameters are chose to satisfy **Theorem 3**. Note that increasing θ shifts the distribution right, *increasing exploration*.

Algorithm 1 Bayesian Optimisation with RGP-UCB

Input: $D_{t_0} = \{x_i, y_i\}_{i=1}^{t_0}$, Γ scale parameter, θ , # Iterations T , Kernel lengthscale l

- 1: **for** $t = t_0$ to T **do**
 - 2: Build a Gaussian Process (GP) with D_t .
 - 3: Set $\kappa_t = \frac{\log(\frac{1}{\sqrt{2\pi}}(t^2+1))}{\log(1+\theta/2)}$ and draw β_t from $\Gamma(\kappa_t, \theta)$
 - 4: Optimise the acquisition function, $\alpha_t(x) = \mu_t(x) + \sqrt{\beta_t \sigma_t(x)}$ to obtain x_{t+1} and use it to sample $y_{t+1} = f(x_{t+1}) + \epsilon_n$ from the system.
 - 5: Augment the data, $D_{t+1} = D_t \cup \{x_{t+1}, y_{t+1}\}$.
 - 6: **end for**
 - 7: **return** $x^+ = x_i$ such that $y_i = \max_{t \in [1, T]} y_t$.
-

3.1 RGP-UCB

While the standard GP-UCB method has a desirable regret bound, it has relatively poor performance. This is due to the β_t used to satisfy this bound being far too large, forcing significant over-exploration. As such, a method for selecting a smaller β_t while maintaining a regret bound is desirable. We show that this can be done by sampling β_t from a distribution, and as such, we call our method *randomised Gaussian process upper confidence bound* (RGP-UCB). Doing so means that we bound the Bayesian regret instead of the regret, but it allows for far greater freedom in selecting β_t , letting it be set far smaller while still maintaining convergence guarantees. However, we require $\beta_t > 0$ since a negative β_t will punish exploration. We also do not want our distribution to suggest a very large β_t as that will cause over-exploration. As such, we draw β_t from a $\Gamma(\kappa_t, \theta)$ distribution. Examples of this distribution can be seen in **Figure 1** and the complete algorithm is shown in **Algorithm 1**. Much like with standard GP-UCB, the parameters of this distribution are chosen to satisfy a regret bound, as per **Theorem 3**. However, we show that we only need to set one of the two distribution parameters for the bound to hold. Unlike the standard GP-UCB, this allows us to tune β_t to substantially improve our methods performance without compromising its theoretical guarantees.

3.2 Theoretical Analysis

Bayesian optimisation is commonly used in high value problems. As such, theoretical measures of its convergence are desirable. The *cumulative regret* is one such measure, and is the cornerstone of GP-UCB [Srinivas *et al.*, 2010]. Regret is simply the difference between the current sampled value and the global optima. The cumulative regret is the sum of the regret over all iterations:

$$R_T = \sum_{t=1}^T [f(x^*) - f(x_t)] \quad (7)$$

As RGP-UCB is probabilistic, we instead need to use the *Bayesian regret* by Russo et al [Russo and Van Roy, 2014]:

$$BR_T = \sum_{t=1}^T \mathbb{E}[f(x^*) - f(x_t)] \quad (8)$$

However, their proof was for Gaussian processes with a finite search space, i.e. $|\mathcal{X}| < \infty$. As such, we follow a method similar to [Srinivas *et al.*, 2010] and [Kandasamy *et al.*, 2017] and introduce a discretisation of our search space into an τ^d grid of equally spaced points, \mathcal{X}_{dis} . We denote $[x]_\tau$ as the closest point to x in \mathcal{X}_{dis} .

With this, we can begin bounding the Bayesian regret of our algorithm by decomposing it into components that are easier to bound.

Lemma 1. *The Bayesian regret of a probabilistic RGP-UCB algorithm, α_{t-1} , over T iterations can be decomposed as*

$$\begin{aligned} BR_T \leq & \underbrace{\sum_{t=1}^T \mathbb{E}[\alpha_{t-1}([x_t]_\tau) - f([x_t]_\tau)]}_{R_1} \\ & + \underbrace{\sum_{t=1}^T \mathbb{E}[f([x^*]_\tau) - \alpha_{t-1}([x^*]_\tau)]}_{R_2} \\ & + \underbrace{\sum_{t=1}^T \mathbb{E}[f(x^*) - f([x^*]_\tau)]}_{R_3} \\ & + \underbrace{\sum_{t=1}^T \mathbb{E}[f([x_t]_\tau) - f(x_t)]}_{R_4} \end{aligned} \quad (9)$$

Proof. This simply follows from the fact that, as $x_t = \arg \max_{x \in \mathcal{X}} \alpha_{t-1}(x)$, we have that $\alpha(x_t) \geq \alpha(x^*)$. \square

With this decomposition, we simply need to find a bound for each term to bound the Bayesian regret. We will start with the second term.

Theorem 1. *Assuming that β_t is drawn from a $\Gamma(\kappa_t, \theta)$ distribution with $\kappa_t = \frac{\log(\frac{1}{\sqrt{2\pi}}(t^2+1))}{\log(1+\theta/2)}$, the following bound holds*

$$R_2 = \sum_{t=1}^T \mathbb{E}[f([x^*]_\tau) - \alpha_{t-1}([x^*]_\tau)] \leq \sum_{t=1}^T \frac{1}{t^2 + 1} \quad (10)$$

Proof. As the posterior distribution of $f(x)$ at iteration $t-1$ is $\mathcal{N}(\mu_{t-1}, \sigma_{t-1}^2)$, the distribution of $f(x) - \alpha_{t-1}(x)|\beta_t$, is simply $\mathcal{N}(-\sqrt{\beta_t}\sigma_{t-1}(x), \sigma_{t-1}^2(x))$. Hence

$$\begin{aligned} \mathbb{E}_\beta [\mathbb{E}_f [\mathbf{1}\{f(x) - \alpha_{t-1}(x) \geq 0\} [f(x) - \alpha_{t-1}(x)|\beta_t]]] \\ \leq \mathbb{E}_\beta \left[\frac{\sigma_{t-1}(x)}{\sqrt{2\pi}} \exp \left\{ -\frac{\sqrt{\beta_t}}{2} \right\} \right] \\ \leq \frac{\sigma_{t-1}(x)}{\sqrt{2\pi}} \mathbb{E}_\beta \left[\exp \left\{ -\frac{\sqrt{\beta_t}}{2} \right\} \right] \end{aligned} \quad (11)$$

We note that the exponential term is simply the moment generating function of $\sqrt{\beta_t}$, which has a closed form for a gamma distribution. This lets us express our inequality as

$$\begin{aligned} \mathbb{E}_\beta [\mathbb{E}_f [\mathbf{1}\{f(x) - \alpha_{t-1}(x) \geq 0\} [f(x) - \alpha_{t-1}(x)|\beta_t]]] \\ \leq \frac{\sigma_{t-1}(x)}{\sqrt{2\pi}} \frac{1}{(1 + \theta_{t-1}/2)^{\kappa_{t-1}}} \end{aligned} \quad (12)$$

As we want this to decay at a sub-linear rate, we need

$$\begin{aligned} \frac{\sigma_{t-1}(x)}{\sqrt{2\pi}} \frac{1}{(1 + \theta_{t-1}/2)^{\kappa_{t-1}}} \leq \frac{1}{(t^2 + 1)|\mathcal{X}_{\text{dis}}|} \\ \kappa_{t-1} \geq \frac{\log \left(\frac{1}{\sqrt{2\pi}} \sigma_{t-1}(x) (t^2 + 1) |\mathcal{X}_{\text{dis}}| \right)}{\log(1 + \theta_{t-1}/2)} \end{aligned} \quad (13)$$

Setting $\kappa_{t-1} = \frac{\log \left(\frac{1}{\sqrt{2\pi}} (t^2 + 1) \right)}{\log(1 + \theta_{t-1}/2)}$ to satisfy this, equation R_2 is bounded by the following:

$$\begin{aligned} R_2 &= \sum_{t=1}^T \mathbb{E} [f([x^*]_\tau) - \alpha_{t-1}([x^*]_\tau)] \\ &\leq \sum_{t=1}^T \sum_{x \in \mathcal{X}} \mathbb{E}_{\beta_t} [\mathbb{E}_f [\mathbf{1}\{f(x) - \alpha_{t-1}(x) \geq 0\} [f(x) - \alpha_{t-1}(x)|\beta_t]]] \\ &\leq \sum_{t=1}^T \frac{1}{t^2 + 1} \end{aligned} \quad (14)$$

□

Next, we attempt to bound the first component.

Theorem 2. Assuming that β_t is drawn from a $\Gamma(k, \theta)$ distribution, the following bound holds

$$\begin{aligned} R_1 &= \sum_{t=1}^T \mathbb{E} [\alpha_{t-1}([x_t]_\tau) - f([x_t]_\tau)] \\ &\leq \sqrt{\left[1 + \frac{k-1}{F^{-1}(1 - \frac{1}{T})} \right] \gamma + F^{-1}\left(1 - \frac{1}{T}\right)} \\ &\quad \times \sqrt{T \sum_{t=1}^T \sigma_{t-1}(x_t)} \end{aligned} \quad (15)$$

Proof. Using Jensen's Inequality we have

$$\begin{aligned} R_1 &= \sum_{t=1}^T \mathbb{E} [\alpha_{t-1}([x_t]_\tau) - f([x_t]_\tau)] \\ &= \sum_{t=1}^T \mathbb{E}_{\beta_t} [\mathbb{E}_f [\alpha_{t-1}([x_t]_\tau) - f([x_t]_\tau)|\beta_t]] \\ &= \sum_{t=1}^T \mathbb{E}_{\beta_t} [\sqrt{\beta_t} \sigma_{t-1}([x_t]_\tau)] \\ &\leq \sum_{t=1}^T \sigma_{t-1}([x_t]_\tau) \sqrt{\mathbb{E}_{\beta_t} [\beta_t]} \end{aligned} \quad (16)$$

We can then use the Cauchy-Schwartz inequality to get

$$\begin{aligned} R_1 &\leq \sqrt{\sum_{t=1}^T \mathbb{E}_{\beta_t} [\beta_t]} \sqrt{\sum_{t=1}^T \sigma_{t-1}^2([x_t]_\tau)} \\ &\leq \sqrt{T \mathbb{E}_{\beta_t} [\max_{t \leq T} \beta_t]} \sqrt{\sum_{t=1}^T \sigma_{t-1}^2([x_t]_\tau)} \end{aligned} \quad (17)$$

As β_t is a gamma distribution with shape parameter κ_t and scale parameter θ , its maximum is given by

$$\mathbb{E} \left[\max_{t \leq T} \beta_t \right] \approx \left[1 + \frac{\kappa_t - 1}{F^{-1}(1 - \frac{1}{T})} \right] \gamma + F^{-1}\left(1 - \frac{1}{T}\right) \quad (18)$$

where γ is the Euler-Mascheroni constant and $F^{-1}(x)$ is the inverse CDF of β . This finally gives us the following bound:

$$\begin{aligned} R_i &\leq \sqrt{\left[1 + \frac{k-1}{F^{-1}(1 - \frac{1}{T})} \right] \gamma + F^{-1}\left(1 - \frac{1}{T}\right)} \\ &\quad \times \sqrt{T \sum_{t=1}^T \sigma_{t-1}^2(x_t)} \end{aligned} \quad (19)$$

□

Finally, we need to bound components R_3 and R_4 . For these, we can use lemma 10 from [Kandasamy *et al.*, 2017].

Lemma 2. At step t , $\forall x \in \mathcal{X}$, $\mathbb{E}[|f(x) - f([x]_\tau)|] \leq \frac{1}{2t^2}$.

This means that we have that

$$R_3 + R_4 \leq 2 \sum_{t=1}^T \frac{1}{2t^2} \leq \frac{\pi^2}{6} \quad (20)$$

With this we can finally find our Bayesian regret bound.

Theorem 3. If β_t is sampled from a $\Gamma(\kappa_t, \theta)$ distribution with

$$\kappa_t = \frac{\log \left(\frac{1}{\sqrt{2\pi}} (t^2 + 1) \right)}{\log(1 + \theta/2)} \quad (21)$$

then the RGP-UCB acquisition function has its Bayesian regret bounded by

$$BR_T \leq \sqrt{\left[1 + \frac{\kappa_t - 1}{F^{-1}\left(1 - \frac{1}{T}\right)}\right] \gamma + F^{-1}\left(1 - \frac{1}{T}\right)} \times \sqrt{T \sum_{t=1}^T \sigma_{t-1}^2(x_t) + \sum_{t=1}^T \frac{1}{t^2 + 1} + \frac{\pi^2}{6}} \quad (22)$$

where γ is the Euler-Mascheroni constant and $F^{-1}(x)$ is the inverse CDF of β_t

Proof. The result follows simply by combining the bounds for the various components. \square

4 Results

In this section we present results that demonstrate the performance of RGP-UCB in comparison to other common acquisition functions. We also demonstrate the impact of varying the θ parameter of the gamma distribution used to sample β_t . The Python code used for this paper can be found at <https://github.com/jmaberk/RGPUCB>.

4.1 Experimental Setup

We test our method against a selection of common acquisition functions on a range of Bayesian optimisations problems. These include a range of synthetic benchmark functions and real-world optimisation problems. These are all transformed into continuous maximisation problems for consistency. In each case, the experiment was run for $40d$ iterations and repeated 10 times with $3d + 1$ different initial points. The initial points are chosen randomly with a Latin hypercube sample scheme [Jones, 2001]. The methods being tested are:

- Our randomised Gaussian process upper confidence bound with $\theta = 8$ (RGP-UCB $\theta = 8$).
- Our randomised Gaussian process upper confidence bound with $\theta = 1$ (RGP-UCB $\theta = 1$).
- Our randomised Gaussian process upper confidence bound with $\theta = 0.5$ (RGP-UCB $\theta = 0.5$).
- Standard Gaussian process upper confidence bound (GP-UCB) [Srinivas *et al.*, 2010].
- Expected improvement (EI) [Jones *et al.*, 1998].
- Thompson sampling [Russo and Van Roy, 2014].
- Max-value entropy search (MES) [Wang and Jegelka, 2017]

Note that we turn all functions that are traditionally minimised into maximisation problems by taking their negative for consistency. As such, higher results are always better.

4.2 Selection of the Trade-off Parameter

An advantage of our method is that it can change its exploration-exploitation balance without compromising its convergence guarantee. This is done by changing the θ parameter in the $\beta_t \sim \Gamma(\kappa_t, \theta)$ distribution. Increasing θ will increase the expected β_t , increasing exploration.

	Dropwave (2D)	Alpine 2 (5D)
$\theta = 0.1$	$0.738 \pm 6.7e-2$	78.9 ± 12.4
$\theta = 0.5$	$0.755 \pm 5.0e-2$	92.1 ± 12.2
$\theta = 1$	$0.754 \pm 6.9e-2$	77.8 ± 12.7
$\theta = 2$	$0.727 \pm 8.6e-2$	77.5 ± 12.5
$\theta = 4$	$0.847 \pm 5.7e-2$	71.5 ± 13.6
$\theta = 8$	$0.848 \pm 3.3e-2$	43.4 ± 9.84
$\theta = 16$	$0.814 \pm 6.2e-2$	45.4 ± 10.0

Figure 2: Best found values using different θ parameters on an exploitation favouring function (Alpine 2) and an exploration favouring function (Dropwave).

As different problems favour different exploration-exploitation balances, we tested a range of θ values on a range of different problems. In **Figure 2**, we show the performance of a range of θ values on an exploitation favouring problem, the Alpine 2 (5D) function, and an exploration favouring problem, the Dropwave (2D) function.

It was found that $\theta = 8$ gives good performance on both the above exploration-favouring problem and other similar problems tested. Likewise, $\theta = 0.5$ is a good choice for exploitation favouring problems. We also note that $\theta = 1$ has decent performance on both problems, making it a good choice for problems where the required exploration-exploitation balance is completely unknown.

4.3 Synthetic Benchmark Functions

The first demonstration of our methods performance is the optimisation of several common synthetic benchmark functions. These are the Dropwave (2D), Sphere (4D), Alpine 2 (5D), and Ackley (5D) functions¹. Results for these are shown in **Figure 3**.

Here we can see that RGP-UCB has competitive performance in all of the above cases. In general, it does significantly better than the standard GP-UCB and the Thompson sampling acquisition functions. EI has better early performance in many cases, as it starts exploiting earlier. However, RGP-UCB tends to have better exploration and therefore often able to beat it in the long-term. MES, on the other hand, has poor early performance in general but catches up to other methods later. However, explorative RGP-UCB ($\theta = 8$) beat it in the exploration favouring dropwave function. In the other synthetic functions, both the balanced ($\theta = 1$) and exploitative ($\theta = 0.5$) RGP-UCB achieved superior results.

The leftmost functions were chosen to be pathological cases which disproportionately favours exploration (Dropwave 2D) and exploitation (Sphere 4D). These can be seen as best-case examples for GP-UCB and EI respectively. However, RGP-UCB is able to out-perform them even on these if its θ parameter is chosen properly, and does so while maintaining its convergence guarantee. This formulation of EI does not have a known regret bound as it follows the standard implementation and hence doesn't satisfy the assumptions required by the current bounds [Bull, 2011; Ryzhov, 2016; Wang and de Freitas, 2014; Nguyen *et al.*, 2017].

¹All benchmark functions use the recommended parameters from <https://www.sfu.ca/~ssurjano/optimization.html>

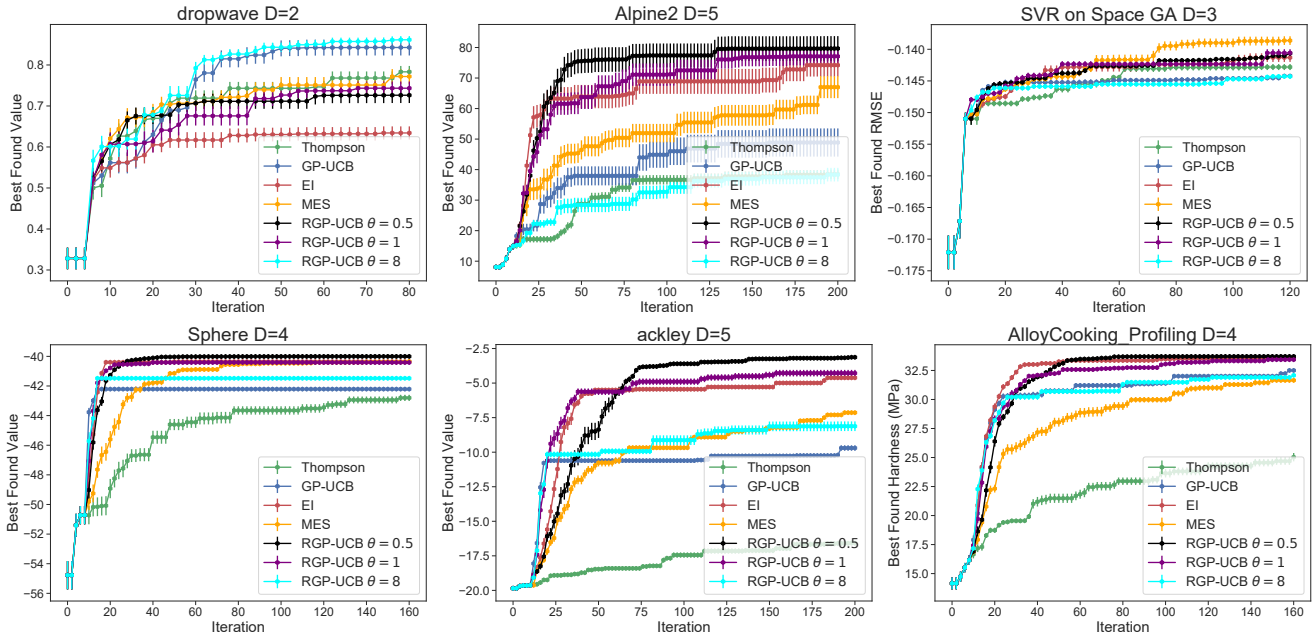


Figure 3: Performance of the RGP-UCB acquisition function compared to other common methods on a range of optimisation problems, including synthetic (left and middle) and real-world (right) examples. The error bars show the standard error.

In the middle two plots, RGP-UCB is superior even with the conservative parametrisation of $\theta = 1$.

4.4 Machine Learning Hyperparameter Tuning

Our first demonstration of the real-world performance of RGP-UCB is the hyperparameter tuning of a support vector regression (SVR) [Drucker *et al.*, 1997] algorithm. This is the support vector machine classification algorithm extended to work on regression problems, with performance measured in root mean squared error (RMSE). It has three hyperparameters, the threshold, ϵ , the kernel parameter, γ , and a soft margin parameter, C . All experiments are done with the public Space GA scale dataset². The results are shown in **Figure 3**.

We can see that, while MES is superior here, the final performance of all three variants of our method exceeds that of standard GP-UCB. The high exploitation and balanced variants are competitive with EI, with the former achieving higher final performance. As with many real-world problems, SVR is known to favour higher exploitation, and is therefore an example of when the user would know to try a smaller θ .

4.5 Materials Science Application: Alloy Heat Treatment

The second demonstration of RGP-UCB’s performance is the optimisation of a Aluminium-Scandium alloy heat treatment simulation [Robson *et al.*, 2003]. The goal of the simulation is to optimise the resulting alloys hardness, measured in MPa. The hardening process is controlled through multiple cooking stages, each with two hyperparameters, the duration and a temperature. As we use a two-stage cooking simulation, there is

a total of four hyperparameters to optimise through Bayesian optimisation. The results are shown in **Figure 3**.

Outside of MES, the results are very similar to the previous SVR example, with the high-exploitation method having the best performance and the balance method being competitive with EI.

4.6 Conclusion

We have developed a modified UCB based acquisition function that has substantially improved performance while maintaining a sub-linear regret bound. We have proved that this bound holds in terms of Bayesian regret while allowing for some flexibility in the selection of its parameters. We have also demonstrated the impact of said parameters on the performance. Moreover, we have shown that its performance is competitive or greater than existing methods in a range of synthetic and real-world applications. In particular, we show that it out-performs the standard GP-UCB implementation on most problems, even when its hyperparameter is chosen conservatively.

Acknowledgements

This research was supported by an Australian Government Research Training Program (RTP) Scholarship awarded to JMA Berk, and was partially funded by the Australian Government through the Australian Research Council (ARC). Prof Venkatesh is the recipient of an ARC Australian Laureate Fellowship (FL170100006).

References

[Brochu *et al.*, 2010] Eric Brochu, Vlad M Cora, and Nando De Freitas. A tutorial on bayesian optimization of expen-

²Dataset can be found at <https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/regression.html>

- sive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.
- [Bull, 2011] Adam D Bull. Convergence rates of efficient global optimization algorithms. *Journal of Machine Learning Research*, 12(Oct):2879–2904, 2011.
- [Drucker *et al.*, 1997] Harris Drucker, Christopher JC Burges, Linda Kaufman, Alex J Smola, and Vladimir Vapnik. Support vector regression machines. In *Advances in neural information processing systems*, pages 155–161, 1997.
- [Gonzalez *et al.*, 2015] Javier Gonzalez, Joseph Longworth, David C James, and Neil D Lawrence. Bayesian optimization for synthetic gene design. *arXiv preprint arXiv:1505.01627*, 2015.
- [Hennig and Schuler, 2012] Philipp Hennig and Christian J Schuler. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13(Jun):1809–1837, 2012.
- [Hernández-Lobato *et al.*, 2014] 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.
- [Jones *et al.*, 1998] Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.
- [Jones, 2001] D. R Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of global optimization*, 21(4):345–383, 2001.
- [Ju *et al.*, 2017] Shenghong Ju, Takuma Shiga, Lei Feng, Zhufeng Hou, Koji Tsuda, and Junichiro Shiomi. Designing nanostructures for phonon transport via bayesian optimization. *Physical Review X*, 7(2):021024, 2017.
- [Kandasamy *et al.*, 2017] Kirthevasan Kandasamy, Akshay Krishnamurthy, Jeff Schneider, and Barnabás Póczos. Asynchronous parallel bayesian optimisation via thompson sampling. *arXiv preprint arXiv:1705.09236*, 2017.
- [Kikuchi *et al.*, 2018] Shun Kikuchi, Hiromi Oda, Shin Kiyohara, and Teruyasu Mizoguchi. Bayesian optimization for efficient determination of metal oxide grain boundary structures. *Physica B: Condensed Matter*, 532:24–28, 2018.
- [Klein *et al.*, 2017] Aaron Klein, Stefan Falkner, Simon Bartels, Philipp Hennig, and Frank Hutter. Fast bayesian optimization of machine learning hyperparameters on large datasets. In *International Conference on Artificial Intelligence and Statistics (AISTATS 2017)*, pages 528–536. PMLR, 2017.
- [Li *et al.*, 2017] Cheng Li, David Rubín de Celis Leal, Santu Rana, Sunil Gupta, Alessandra Sutti, Stewart Greenhill, Teo Slezak, Murray Height, and Svetha Venkatesh. Rapid bayesian optimisation for synthesis of short polymer fiber materials. *Scientific reports*, 7(1):1–10, 2017.
- [Nguyen *et al.*, 2017] Vu Nguyen, Sunil Gupta, Santu Rana, Cheng Li, and Svetha Venkatesh. Regret for expected improvement over the best-observed value and stopping condition. In *Asian Conference on Machine Learning*, pages 279–294, 2017.
- [Rasmussen and Williams, 2006] Carl Edward Rasmussen and Christopher KI Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.
- [Robson *et al.*, 2003] JD Robson, MJ Jones, and PB Prangnell. Extension of the n-model to predict competing homogeneous and heterogeneous precipitation in al-sc alloys. *Acta Materialia*, 51(5):1453–1468, 2003.
- [Russo and Van Roy, 2014] Daniel Russo and Benjamin Van Roy. Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 39(4):1221–1243, 2014.
- [Ryzhov, 2016] Ilya O Ryzhov. On the convergence rates of expected improvement methods. *Operations Research*, 64(6):1515–1528, 2016.
- [Scott *et al.*, 2011] Warren Scott, Peter Frazier, and Warren Powell. The correlated knowledge gradient for simulation optimization of continuous parameters using gaussian process regression. *SIAM Journal on Optimization*, 21(3):996–1026, 2011.
- [Shahriari *et al.*, 2015] 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, 2015.
- [Snoek *et al.*, 2012] 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.
- [Srinivas *et al.*, 2010] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: no regret and experimental design. In *Proceedings of the 27th International Conference on Machine Learning*, pages 1015–1022, 2010.
- [Turgeon *et al.*, 2016] Martine Turgeon, Cindy Lustig, and Warren H Meck. Cognitive aging and time perception: roles of bayesian optimization and degeneracy. *Frontiers in aging neuroscience*, 8:102, 2016.
- [Wang and de Freitas, 2014] Ziyu Wang and Nando de Freitas. Theoretical analysis of bayesian optimisation with unknown gaussian process hyper-parameters. *NIPS Workshop on Bayesian Optimization*, 2014.
- [Wang and Jegelka, 2017] Zi Wang and Stefanie Jegelka. Max-value entropy search for efficient bayesian optimization. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3627–3635. JMLR. org, 2017.
- [Xia *et al.*, 2017] Yufei Xia, Chuanzhe Liu, YuYing Li, and Nana Liu. A boosted decision tree approach using bayesian hyper-parameter optimization for credit scoring. *Expert Systems with Applications*, 78:225–241, 2017.