High Dimensional Bayesian Optimization via Restricted Projection Pursuit Models

Chun-Liang Li  Kirthevasan Kandasamy  Barnabás Póczos  Jeff Schneider
{chunlial, kandasamy, bapoczos, schneide}@cs.cmu.edu
Carnegie Mellon University

Abstract

Bayesian Optimization (BO) is commonly used to optimize blackbox objective functions which are expensive to evaluate. A common approach is based on using Gaussian Process (GP) to model the objective function. Applying GP to higher dimensional settings is generally difficult due to the curse of dimensionality for nonparametric regression. Existing works make strong assumptions such as the function is low-dimensional embedding (Wang et al., 2013) or is axis-aligned additive (Kandasamy et al., 2015). In this paper, we generalize the existing assumption to a projected-additive assumption. Our generalization provides the benefits of i) greatly increasing the space of functions that can be modeled by our approach, which covers the previous works (Wang et al., 2013; Kandasamy et al., 2015) as special cases, and ii) efficiently handling the learning in a larger model space. We prove that the regret for projected-additive functions has only linear dependence on the number of dimensions in this general setting. Directly using projected-additive GP (Gilboa et al., 2013) to BO results in a non-box constraint, which is not easy to optimize. We tackle this problem by proposing a restricted-projection-pursuit GP for BO. We conduct experiments on synthetic examples and scientific and hyper-parameter tuning tasks in many cases. Our method outperforms existing approaches even when the function does not meet the projected additive assumption. Last, we study the validity of the additive and projected-additive assumption in practice.

1 Introduction

Many applications in machine learning, science and engineering can be treated as zeroth-order optimization of a smooth function which is not analytically available and is usually expensive to evaluate. By querying the underlying objective function \( f \), the goal is to find those points that have maximal function values.

Bayesian Optimization is a popular method to solve zeroth-order optimization. A commonly-used approach is modeling the unknown objective function \( f \) with a Gaussian Process (GP) (Mockus and Mockus, 1991). At time \( t \), BO estimates the posterior GP from queried points and constructs an acquisition function \( \varphi_t \) using upper confidence bound, Thompson sampling, or other heuristics (Brochu et al., 2010). Then BO maximizes \( \varphi_t \) to determine \( x_t \), the next point to be queried. Acquisition functions such as upper confidence bound also apply to the bandit setting and GP assumptions. Gaussian process bandits and Bayesian optimization (GPB/BO) have been successfully applied in many applications (Brochu et al., 2010). Besides Gaussian Process, other models including random forests (Hutter et al., 2011) and deep neural network (Snoek et al., 2015) are also applied to BO. However, neither of them has the regret analysis with asymptotic results.

Although there are many low dimensional applications (Wang et al., 2013) where existing algorithms work well (typically dimension < 10), there are also important high dimensional applications (Yamins et al., 2013; Gonzalez et al., 2014) for which satisfying high dimensional optimization methods haven not been developed yet. High-dimensional BO is generally non-trivial on both the statistical and computational sides. On the statistical side, GPB/BO is exponentially difficult in high dimensions with regard to query complexity (Srinivas et al., 2010). In addition, there is the computational cost of optimizing \( \varphi_t \). Commonly used algorithms require computational time that is exponential in dimension. BO literature typically assumes
that the cost for maximising $\varphi_t$ is negligible in comparison to the cost of evaluation of $f$. While this might be true in some cases, in several cases we are constrained in time to optimize $\varphi_t$, such as the parameter tuning in machine learning applications (Snook et al., 2012).

To address the difficulty of high dimensionality, the low-dimensional assumption (Chen et al., 2012; Wang et al., 2013; Djolonga et al., 2013) assumes the function only varies within a low-dimensional subspace. On the other hand, Kandasamy et al. (2015) make the additive assumption by assuming the function $f$ is a sum of functions of small, disjoint groups of dimensions. The sum is high-dimensional while each term of the sum is low-dimensional. The former approach limits a high-dimensional function in a low dimensional embedding, while the latter is restricted to an axis-aligned representation. We seek to get the best of both of these methods without their corresponding drawbacks.

In this paper, we make the following contributions. First, we generalize the existing two assumptions to a projected-additive assumption (Hastie and Tibshirani, 1990) to handle a broader class of functions. The additive assumption and the low-dimensional assumption are special cases under the projected-additive assumption. We show the regret has only linear dependence on the dimension $D$ when $f$ is under the project-additive assumption (Section 4). Second, although Gilboa et al. (2013) study GPs with a projected-additive approximation, it is non-trivial how to directly apply Gilboa et al. (2013) to BO in practice. After projection, the size of the distorted function domain could be exponentially increased which causes difficulty for optimizing the acquisition functions. We propose the restricted-projection-pursuit algorithm to solve this difficulty (Section 4.1) with good empirical performance on both synthetic and real-world data (Section 5). Last, we also study the case when $f$ does not meet these assumptions but still work in practice with theoretical explanations (Section 6).

2 Preliminaries

We aim to maximize a function $f : \mathcal{X} \rightarrow \mathbb{R}$ where $\mathcal{X}$ is a rectangular region in $\mathbb{R}^D$. Without loss of generality, we assume $\mathcal{X} = [0, 1]^D$. We also assume that we are not allowed to know the exact function formulation but only know the function evaluation by querying at $x \in \mathcal{X}$ and obtain a noisy observation $y = f(x) + \epsilon$, where $\epsilon$ is Gaussian white noise.

We denote the optimal point as $x^* = \arg\max_{x \in \mathcal{X}} f(x)$. At time $t$, we query point $x_t$ and incur the instantaneous regret $r_t = f(x_t) - f(x^*)$. In the bandit setting, we are interested in minimizing the cumulative regret $R_T = \sum_{t=1}^T r_t = \sum_{t=1}^T (f(x_t) - f(x^*))$, where $T$ is the total number of iterations. In the optimization setting, we are interested in minimizing the simple regret $S_T = \min_{x \leq T} r_1 = f(x_1) - \max_{x_t} f(x_t)$. Note that $S_T \leq \frac{1}{T} R_T$, so any algorithm with asymptotic sublinear cumulative regret also has vanishing simple regret.

In high-dimensional optimization problems, usually we assume some smoothness conditions on $f$ to make the problem tractable. We assume $f$ is sampled from a zero mean Gaussian Process (Rasmussen and Williams, 2006) with a covariance kernel $\kappa : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ and $\epsilon \sim N(0, \eta^2)$. Two commonly used kernels are the squared exponential (SE) and Matérn kernels (Rasmussen and Williams, 2006).

At each time $t$, we construct the acquisition function $\varphi_t$ and maximize it to determine which point $x_t$ to query. A common acquisition function used in bandit problems is the Upper Confidence Bound (UCB) (Cox and John, 1997). In this work, we focus on using the Gaussian Process Upper Confidence Bound (GP-UCB) (Srinivas et al., 2010) to construct $\varphi_t$. That is, $\varphi_t(x) = \mu_t(x) + \beta_t^{1/2} \sigma_t^{-1}(x)$, and $x_t = \arg\max_x \varphi_t(x)$. Here, $\mu_t$ is the posterior mean after $t$ points, $\sigma_t^{-1}$ is the posterior standard deviation and $\beta_t^{1/2}$ is a coefficient to negotiate the tradeoff between exploration and exploitation. We follow Brochu et al. (2010) to use the Dividing Rectangles (DiRect) algorithm (Jones et al., 1993) to optimize the acquisition function. The time complexity of DiRect is $O(\epsilon^{-D})$ if we want to achieve $\epsilon$ accuracy, which makes high dimensional BO challenging given the limited computational time in practice.

3 Additive Gaussian Process Models

To deal with high dimensional problems in the GP framework, Duvenaud et al. (2011) use an additive kernel. Kandasamy et al. (2015) use a group-additive kernel for BO by assuming independence between groups of dimensions. They demonstrate that it achieves significant performance gains in high dimensions. In this section, we briefly review the group-additive model of Kandasamy et al. (2015) and abuse the word “additive” to mean “group-additive” for simplicity.

Assume the function $f$ can be decomposed into the following group-additive form,

$$f(x) = f^{(1)}(x^{(1)}) + \ldots + f^{(M)}(x^{(M)}),$$

and the mean and kernel functions can be decomposed as well:

$$\mu(x) = \mu^{(1)}(x^{(1)}) + \ldots + \mu^{(M)}(x^{(M)})$$

$$\kappa(x, x') = \kappa^{(1)}(x^{(1)}, x'^{(1)}) + \ldots + \kappa^{(M)}(x^{(M)}, x'^{(M)}),$$
where $x^{(j)} \in \mathcal{X}^{(j)} = [0,1]^d$ are disjoint lower dimensional components. For simplicity, assume $d_j = d$ for all $j$, and thus $D = Md$.

Inference in Additive GPs: By (1), at time $t$, given the noisy labels $y_t = \{y_1, \ldots, y_t\}$ at points $X_t = \{x_1, \ldots, x_t\}$ and a query point $x$, infer the posterior distribution of $f(x)$ by calculating the posterior distributions of $\tilde{f}^{(j)}(x)$ individually. For each $\tilde{f}^{(j)}$, the joint distribution can be written as

$$
(\tilde{f}^{(j)} | y_t) \sim \mathcal{N}(0, \begin{bmatrix} \kappa(x^{(j)}, x^{(j)}) & \kappa(x^{(j)}, x_t^{(j)}) \\ \kappa(x_t^{(j)}, x^{(j)}) & \kappa(x_t, x_t^{(j)}) + \eta^2 I_d \end{bmatrix}),
$$

where the $p^{th}$ element of $\kappa(x_t^{(j)}, x^{(j)}) \in \mathbb{R}^d$ is $\kappa(x_t^{(j)}, x^{(j)})$ and the $(p,q)^{th}$ element of $\kappa(x_t, x_t^{(j)}) \in \mathbb{R}^d \times \mathbb{R}^d$ is $\kappa(x_t, x_t^{(j)})$.

Now, the posterior distribution $\mathcal{N}(\mu(x^{(j)}), \sigma(x^{(j)}))$ of $\tilde{f}^{(j)}$ is,

$$
\tilde{f}^{(j)} | x, X_t, y_t \sim \mathcal{N}(\kappa(x^{(j)}, x^{(j)})\Delta^{-1}y_t, \kappa(x^{(j)}, x^{(j)}) - \kappa(x^{(j)}, x_t^{(j)})\Delta^{-1}\kappa(x_t, x^{(j)})),
$$

where $\Delta = \kappa(x_t, x_t) + \eta^2 I_d$.

Additive Acquisition Function: Using the inferred posterior distribution of $f^{(j)}(x^{(j)})$, define the Additive Gaussian Process Upper Confidence Bound (Add-GP-UCB) to be

$$
\tilde{\varphi}_t(x) = \mu_{t-1}(x) + \beta_t^{1/2} \sum_{j=1}^{M} \sigma_{t-1}^{(j)}(x^{(j)})
= \sum_{j=1}^{M} \mu_{t-1}^{(j)}(x^{(j)}) + \beta_t^{1/2} \sigma_{t-1}^{(j)}(x^{(j))).
$$

Then the optimization of the acquisition function $\tilde{\varphi}_t(x)$ can be decomposed as

$$
x_t^{(j)} = \arg\max_{x^{(j)} \in \mathcal{X}^{(j)}} \mu_{t-1}^{(j)}(x^{(j)}) + \beta_t^{1/2} \sigma_{t-1}^{(j)}(x^{(j)}).
$$

To achieve $(\epsilon, \delta)$ accuracy, the time complexity of optimizing the Add-GP-UCB acquisition function is $O(M\epsilon^{-2d})$, which is more efficient than $O(\epsilon^{-2D})$ of the original UCB algorithm.

The algorithm is shown in Algorithm 1. Note that Kandasamy et al. (2015) learn the decomposition and hyperparameters every $N_{cyc}$ iterations to make the algorithm more efficient.

Algorithm 1 Add-GP-UCB

**Input:** $N_{init}$, $N_{cyc}$, $d$, $M$

- $D_0 \leftarrow$ Uniformly sample $N_{init}$ points from $\mathcal{X}$.
- for $t = 1, 2, \ldots$
  - 1. if $(t \mod N_{cyc} = 1)$, Learn the Kernel hyper parameters and the decomposition $\{X_j\}$ by maximising the GP marginal likelihood.
  - 2. Get $x_t^{(j)}$ via (5) for all $j$.
  - 3. Get $y_t$ via querying $f$ at $x_t$.
  - 4. $D_t = D_{t-1} \cup \{x_t, y_t\}$.

4 Projection Pursuit Model

In nonparametric regression, the projected-additive assumption (Hastie and Tibshirani, 1990) is used to generalize the additive assumption. Here, we generalize the projective-additive model used in Gilboa et al. (2013) to a group version as follows,

$$
f(x) = f^{(1)}(W^{(1)}x) + \ldots + f^{(M)}(W^{(M)}x),
\mu(x) = \mu^{(1)}(W^{(1)}x) + \ldots + \mu^{(M)}(W^{(M)}x), \kappa(x, x') = \kappa^{(1)}(W^{(1)}x, (W^{(M)}x)x') + \ldots + \kappa^{(M)}((W^{(M)}x)x', (W^{(M)}x)x').
$$

Let $W$ denote the collection of $W^{(j)} \in \mathbb{R}^{D \times d}$. We assume $W \in \mathbb{R}^{D \times d}$. $W$ is invertible and unknown to us. In the following, we will use “projected-additive” to denote the model in (6).

The proposed group-projected-additive model is a generalization of several previously proposed models. If $W = I$, the model is the additive model proposed in Kandasamy et al. (2015). Also, the proposed assumption cover the low-dimensional assumption of Wang et al. (2013) as well.

Algorithm 2 PP-GP-UCB

**Input:** $N_{init}$, $N_{cyc}$, $d$, $M$

- $D_0 \leftarrow$ Uniformly sample $N_{init}$ points from $\mathcal{X}$.
- for $t = 1, 2, \ldots$
  - 1. Learn $W$ on $D_{t-1}$ by Gilboa et al. (2013) and get the projected data $Z_{t-1}$.
  - 2. Perform Step 1-2 in Algorithm 1 on $Z_t$ to get $z_t^{(j)}$ for all $j$ and $x_t = (W^{-1})z_t$.
  - 3. Get $y_t$ via querying $f$ at $x_t$.
  - 4. $D_t = D_{t-1} \cup \{x_t, y_t\}$.

Gilboa et al. (2013) proposed an efficient algorithm called projection pursuit GP regression (PP-GP) to approximate $W$. It partitions $W$ into several vectors $w_j$ and learn them individually via the state-space model using Expectation Maximization. The algorithm utilizes the approximation inference with iteration complexity that is linear time in number of data.
In BO, the number of queries is usually limited, which makes applying PP-GP feasible in practice. Observe that the projection pursuit GP regression is equivalent to applying the additive model in Section 3 on the projected data $Z = XW$ (Gilboa et al., 2013). Our proposed approach, the PP-GP-UCB algorithm for Bayesian Optimization, is described in Algorithm 2, and we also provide a regret bound about its performance in Theorem 2. Due to the space limits, we defer all proofs to the Appendix. First we define the Maximum Information Gain (Srinivas et al., 2010).

**Definition 1.** (Maximum information gain) Let $f \sim \mathcal{GP}(0, \kappa)$ and $y_i = f(x_i) + \epsilon_i$, where $\epsilon_i \sim \mathcal{N}(0, \eta^2)$. Given $A = \{x_1, \ldots, x_T\} \subset \mathcal{X}$, let $f_A = \{f(x_i)\}$, $y_A = f_A + \epsilon_A$ and $I$ be the mutual information. The maximum information gain $\gamma_T$ after $T$ round is

$$\gamma_T = \max_{A \in \mathcal{X}, |A| = T} I(y_A; f_A).$$

$\gamma_T$ characterizes the difficulty of the problem and captures the difficulty with dimensionality. For the SE kernel, $\gamma_T \in O((D \log T)^{D+1})$. If it is a sum of SE kernels, Kandasamy et al. (2015) show that it improves to $O(Dd^d (\log T)^{D+1})$.

**Theorem 2.** Suppose $g(z) = f(x)$, where $z = W^T x$ and $x$ is constructed by sampling $g^{(j)} \sim \mathcal{GP}(0, \kappa^{(j)})$ for $j = 1, \ldots, M$ and then summing them as (6). Suppose also that we maximize the acquisition function $\tilde{\phi}_t(z) = \sum_{j=1}^M \alpha_j (z^{(j)}) + \beta_t / 2 \sigma_i^{-1} (z^{(j)})$ to within $O(t^{-1/2})$ accuracy at time step $t$ with $\beta_t = \tilde{O}(d \log t)$. Under these conditions PP-GP-UCB attains simple regret $S_T \in \tilde{O}(\sqrt{Dd^d \log T / T})$ with high probability, where $\gamma_T = O((Dd^d (\log T)^{d+1})$ for SE kernel.

### 4.1 Restricted Projection

In BO, we maximize a function within a box region $\mathcal{X} = [0, 1]^D$. The box constraints benefit the additive model because we can optimize the acquisition function of each group $x^{(i)}$ independently under the box constraints $x^{(j)} \in [0, 1]^{d_j}$ without considering other groups of dimensions.

![Figure 1: Two-dimensional example of transformed domains.](image1)

When we apply PP-GP-UCB to BO, the constraints of the projected data $z$ is $((W^T)^{-1}z) \in [0, 1]^D$, which is not a box constraint. We then restrict the range of $z^{(i)}$ to $[0, 1]^{d_i} \subset [0, 1]^D$. We then define the restricted projection matrix as

$$W = (1 - \alpha)W + \alpha I,$$

where $\alpha \in [0, 1]$. If $\alpha = 1$, it is the projection-pursuit GP regression; if $\alpha = 0$, it is the projection-pursuit GP regression. From the

![Figure 2: Two-dimensional example of the different constraints](image2)
geometric perspective, we optimize the marginal likelihood while consider the ratio of the size of the outer box to determine $\alpha$.

$$\arg\max_{\alpha} \log P(y_t|X_t; \hat{W})$$

subject to

$$\frac{\text{outer}(X|W)}{|X|} \leq 1 + \delta$$

(8)

$$\hat{W} = (1 - \alpha)W + \alpha I,$$

where $|X| = |W|$ is the size (volume) of the transformed domain, and $\text{outer}(X|W) = \prod_{d=1}^{D} |w_d|$ is the size of the outer-box domain, where $w_d$ is the $d^{th}$ column of $W$. By controlling $\delta$, we could restrict the size of searching space when we optimize the acquisition function. We call the constraint $\frac{\text{outer}(X|W)}{|X|} \leq 1 + \delta$ as $\delta$-ratio constraint. The example of the restricted projection is shown in Figure 2(c). Then we only need to optimize one single variable $\alpha$ to control the increased domain, which is simpler than solving $W$ with $\delta$-ratio constraint directly. In practice, we observe the grid search by checking the constraint could bring us satisfactory empirical performance.

We call the proposed algorithm Restricted-Projection-Pursuit GP-UCB (RPP-GP-UCB). The pseudo-code of the algorithm is shown in Algorithm 3.

Algorithm 3 RPP-GP-UCB

Input: $N_{init}$, $N_{gc}$, $d$, $M$, $\delta$

• $D_0 \leftarrow$ Uniformly sample $N_{init}$ points from $\mathcal{X}$.

• for $t = 1, 2, \ldots$

1. if $(t \mod N_{gc}) = 1$ Learn $\hat{W}$ with $\delta$-ratio constraint with warm start to make this step more efficient and get the projected data $Z_t$.

2. Perform Step 1-2 in Algorithm 1 on $Z_t$ with the domain $\text{outer}(W)$ to get $z^{(j)}_t$ for all $j$ and $x_t = (\hat{W}^{-1})^T z_t$.

3. Get $y_t$ via querying $f$ at $x_t$.

4. $D_t = D_{t-1} \cup \{(x_t, y_t)\}$.

5.5 Synthetic Data

First we study the synthetic data to understand the behavior of the proposed algorithm. We generate the objective function $f$ as follows. Let

$$f_d(z) = \log (0.1 \times \text{mvnpdf}(z, v_1, \sigma^2) + 0.1 \times \text{mvnpdf}(z, v_2, \sigma^2) + 0.8 \times \text{mvnpdf}(z, v_3, \sigma^2)),$$

where $\sigma^2 = 0.01d^{0.1}$ and mvnpdf($x$, $\mu$, $\sigma^2$) is the probability density at $x$ of the multivariate Gaussian distribution with mean $\mu$ and covariance function $\sigma^2 I$. $v_1, v_2, v_3$ are fixed points in $\mathbb{R}^d$.

We then define $f(x) = f_d(A_1^T x) + \cdots + f_d(A_D^T x)$, where $A_i \in \mathbb{R}^{D \times d}$, $x \in \mathbb{R}^D$ and $D = Md'$. We denote $A \in \mathbb{R}^{D \times D}$ as the collection of $A_i$.

We study the cases when $D = 50$ and $D = 100$, with $d' = D/2$ and $\delta = 0.1$. We then set $A = I + S$ and sample each element in $S$ from $\text{Unif}(-1/4, -1/4)$ and $\text{Unif}(-1/8, -1/8)$ for $D = 50$ and $D = 100$, respectively to make $A$ invertible. Also, we experiment with ADD-GP-UCB and RPP-GP-UCB for the following values of group dimensionality $d = (5, 10, 25)$. Finally, we report the average simple regret of ADD-GP-UCB (ADD-$d$) and RPP-GP-UCB (RPP-$d$) with standard errors in Table 1 and 2. More experiment results for other algorithms (Wang et al., 2013; Hutter et al., 2011) can be found in Appendix.

From Table 1, for $D = 50$, neither ADD-GP-UCB nor RPP-GP-UCB are statistically expressive when we use $d = 5$. When we increase $d$ to 10 and 25, both ADD-GP-UCB and RPP-GP-UCB outperform GP-UCB . Also, we could notice that $d = 10$ outperforms $d = 25$. Although we set $d = 25$, which is equal to $d'$, optimizing in 25 dimensions is more difficult than optimizing in 10 dimensions. The imperfect optimization of $\varphi_t$ and the difficulty of estimating functions in higher dimensions degrade the performance.

In Table 2, we have similar observations for $D = 100$. It is worth noticing that optimizing acquisition function gets more difficult under higher dimensional setting. When $D = 100$, both ADD-GP-UCB and RPP-GP-UCB perform better than GP-UCB for all values of $d$. 

1Note that the original domain is $[0, 1]^D$. 

5 Experiment

To make the experiment realistic and demonstrate the efficacy of RPP-GP-UCB, we optimize the acquisition function under limited budget with DiRect as used in Brochu et al. (2010). For RPP-GP-UCB, we use Gilboa et al. (2013) to find $W$, and find $\alpha$ in (7) by grid search in the range feasible to the constraint. For grouping the dimensions in Add-GP-UCB and RPP-GP-UCB, we create the $M$ combinations randomly and choose the one maximizing the marginal likelihood, where $M$ is the number of groups. For all GPB/BO-based algorithms, we set $N_{init} = 10$, $N_{gc} = 25$. In all experiments, we use SE kernels and maximize the marginal likelihood $P(y_t|X_t; \theta)$ to find the hyperparameters $\theta$. For additive and projected-additive kernels, we use the same hyperparameters for all the kernels. We run each algorithm 50 times for each dataset with different initializations and report the average results.
Table 1: The simple regret results of the synthetic data with $D = 50$.

<table>
<thead>
<tr>
<th></th>
<th>GP-UCB</th>
<th>ADD-5</th>
<th>RPP-5</th>
<th>ADD-10</th>
<th>RPP-10</th>
<th>ADD-25</th>
<th>RPP-25</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T = 500$</td>
<td>103.9 ± 1.9</td>
<td>180.3 ± 10.1</td>
<td>151.1 ± 9.7</td>
<td>35.4 ± 2.2</td>
<td>30.2 ± 1.7</td>
<td>164.5 ± 9.4</td>
<td>152.4 ± 6.5</td>
</tr>
<tr>
<td>$T = 1000$</td>
<td>91.4 ± 1.3</td>
<td>153.4 ± 7.1</td>
<td>136.8 ± 8.0</td>
<td>31.3 ± 2.1</td>
<td>25.1 ± 1.9</td>
<td>77.8 ± 3.4</td>
<td>75.7 ± 2.4</td>
</tr>
</tbody>
</table>

Table 2: The simple regret results of the synthetic data with $D = 100$.

<table>
<thead>
<tr>
<th></th>
<th>GP-UCB</th>
<th>ADD-5</th>
<th>RPP-5</th>
<th>ADD-10</th>
<th>RPP-10</th>
<th>ADD-25</th>
<th>RPP-25</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T = 500$</td>
<td>114.1 ± 1.0</td>
<td>168.0 ± 8.1</td>
<td>150.6 ± 12.2</td>
<td>109.8 ± 3.6</td>
<td>98.9 ± 2.6</td>
<td>175.8 ± 3.7</td>
<td>159.0 ± 2.6</td>
</tr>
<tr>
<td>$T = 1000$</td>
<td>106.6 ± 0.9</td>
<td>92.6 ± 3.0</td>
<td>89.5 ± 5.0</td>
<td>41.5 ± 1.4</td>
<td>34.5 ± 1.1</td>
<td>85.8 ± 1.7</td>
<td>85.1 ± 1.5</td>
</tr>
</tbody>
</table>

To study whether RPP-GP-UCB is significantly better than Add-GP-UCB, we conduct the $t$-test at 95% confidence level. The results are given in Tables 3 and 4. Note that when $A$ is diagonal, $f$ is additive. Since $A$ is similar to a diagonal matrix but with small perturbations, we could still approximate the function well. Therefore, it is not surprising that the performance difference is not significant when we use large $d$ (e.g., $d = 25$). However, in terms of the average performance, RPP-GP-UCB still brings us the performance gain under this situation.

<table>
<thead>
<tr>
<th>$d = 5$</th>
<th>$d = 10$</th>
<th>$d = 25$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T = 500$</td>
<td>win</td>
<td>win</td>
</tr>
<tr>
<td>$T = 1000$</td>
<td>win</td>
<td>win</td>
</tr>
</tbody>
</table>

Table 3: RPP-GP-UCB versus Add-GP-UCB based on $t$-test at 95% confidence level when $D = 50$. "win" means that RPP-GP-UCB is better than Add-GP-UCB at the 95% significance level.

<table>
<thead>
<tr>
<th>$d = 5$</th>
<th>$d = 10$</th>
<th>$d = 25$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T = 500$</td>
<td>tie</td>
<td>win</td>
</tr>
<tr>
<td>$T = 1000$</td>
<td>tie</td>
<td>win</td>
</tr>
</tbody>
</table>

Table 4: RPP-GP-UCB versus Add-GP-UCB based on $t$-test at 95% confidence level when $D = 100$.

Without any restriction on $W$ ($\delta = \infty$), it is equivalent to directly applying Gilboa et al. (2013) on BO. However, the average simple regret is more than 500 when $D = 50$ and $D = 100$. We address the bad performance to two reasons. First, without any restriction, the size of the outer box is much larger, which causes the algorithm query many points out of the original domain. Second, we observe that $W$ tends to overfit the queried points due to the insufficient queries.

5.1.1 Study of $\delta$

Finally we study the trade-offs in choosing $\delta$. In Table 5, larger $\delta$ results in better performance because it is more statistically expressive. If we are allowed to use more DiRect evaluations, we could expect larger $\delta$ could further improve the performance. When $d$ is large enough, such as $d = 25$, even the small $\delta$ could be statistically expressive. So using larger $\delta$ is less efficient because the algorithm searches a larger domain.

<table>
<thead>
<tr>
<th></th>
<th>$d = 5$</th>
<th>$d = 10$</th>
<th>$d = 25$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\delta = 0$ (ADD)</td>
<td>53.4 ± 7.1</td>
<td>31.3 ± 2.1</td>
<td>77.8 ± 3.4</td>
</tr>
<tr>
<td>$\delta = 0.05$</td>
<td>144.8 ± 9.7</td>
<td>27.9 ± 1.8</td>
<td>73.3 ± 3.6</td>
</tr>
<tr>
<td>$\delta = 0.1$</td>
<td>136.8 ± 9.7</td>
<td>25.1 ± 1.9</td>
<td>75.7 ± 2.4</td>
</tr>
</tbody>
</table>

Table 5: The simple regret of RPP-GP-UCB when $D = 50$ and $T = 1000$.

5.2 Real-world Data

Here we consider two real-world tasks, an astrophysical simulation using data from the Sloan Digital Sky Survey (SDSS) (Tegmark et al., 2006) and face detection using the Viola & Jones Cascade classifier (VJ) (Viola and Jones, 2001). The software of SDSS computes the likelihood of a simulation based astrophysical model with cosmological parameters and spectrum data. Higher likelihood could help cosmologists understand the links between these parameters and the power spectra. Therefore, we wish to find the parameters to maximize the likelihood value. The software uses 9 parameters. To emulate the real-world setting, we augment it to 40 dimensions with dummy variables and perturb the parameters with small correlations by $L_{40 \times 40} + S$, where $S$ is sampled from $\text{Unif}(-0.1, 0.1)$. On the other hand, VJ is a machine learning parameter tuning task for detecting whether an image contains a face. The sampled faces are shown in Figure 3. VJ uses $K$ weak classifiers with $K$ thresholds. If the scores from $K$ classifiers of images pass $K$ thresholds, it is classified as positive. We use the implementation in OpenCV (Bradski and Kaehler, 2008) with $K = 22$ parameters without augmentation.

Figure 3: The sample faces and non faces of VJ data.

Note that RPP-GP-UCB is still better than Add-GP-UCB even though we use smaller $\delta$. It demonstrates the validity of the proposed algorithm.
set $\delta = 0.05$. The average results with standard errors are shown in Figure 4, where ADD/RPP-\textit{d} means each group has $d$ dimensions and REMBO-d assumes the number of low dimensions is $d$.

In practice, the correlation between different dimensions may be larger than the synthetic data used in Section 5.1, then the additive model may not have satisfactory performance by loosing some more information. According to Figure 4, we could observe RPP-GP-UCB with small $\delta$ significantly outperforms Add-GP-UCB when they use the same $d$. Even when we use $d = 1$ for RPP-GP-UCB, it could result in competitive or better performance than Add-GP-UCB with larger $d$, which justifies RPP-GP-UCB is more statistically expressive than other models. On the other hand, REMBO does not have satisfactory performance. For SDSS, even though $D/4$ is larger than the number of the underlying low dimensions (9 dimensions). We address the bad performance into the imperfect optimization for acquisition functions, and the empirical choice of the search domain used in Wang et al. (2013) is too small to include the optimal point. For VJ, since the underlying low dimensional assumption does not hold, optimizing $D/4$ dimensions is not enough to get satisfactory performance.

6 Discussion

The additive assumption is usually non-realistic. However, Kandasamy et al. (2015) show that it still works well in practice without a theoretical explanation. In this section, we aim to provide some theoretical insights to understand when the projected-additive assumption works even though the objective function does not meet the additive assumption. For simplicity, we assume the number of dimension of each lower dimensional group $d_j = 1$ for all $j$, but one can easily generalize the following results to $d_j > 1$.

Assumption 3. Let $f : \mathcal{X} \rightarrow \mathbb{R}$, where $\mathcal{X} = [0, 1]^D$. Define $f(z) = f(\sqrt{D}z)$. The domain $\mathcal{Z}$ of $f$ is $[0, 1/\sqrt{D}]^D$, and $||z|| = \epsilon$, where $\epsilon < 1$. Suppose that $f$ has bounded second-order and third-order derivatives.

Proposition 4. Under Assumption 3, there exists an additive approximation $g(1)$ modeling the first order derivative such that $\max_{x} |f(x) - g(x)| = O(\epsilon^2)$.

Proposition 5. Under Assumption 3, there exists a projected-additive approximation $h(6)$ modeling the first and second order derivatives such that $|f(z) - h(z)| = O(\epsilon^3)$.

Even though $f(x)$ is not additive, if $f(x)$ is not too wiggly and has bounded high-order derivatives, which is a common assumption in nonparametric regression, then the additive model could be treated as a first-order approximation and the projected-additive model as a second-order approximation.

As we show in Section 5.1, since $A$ is similar to an identity matrix, the correlation between dimensions is small. Therefore, by only utilizing the first-order approximation (Add-GP-UCB) could still result in satisfactory performance. However, the correlation between different dimensions may be larger than the used synthetic data, then we could see RPP-GP-UCB significantly outperforms Add-GP-UCB in Figure 4 by modeling the second-order information.

Next, we show that the difference between the suboptimality that arises through the approximation of the function and the global optimal is bounded.

Assumption 6. The posterior mean function is defined as $\mu_t(x) = \kappa(x, X_t) (\kappa(X_t, X_t) + \eta I)^{-1} y_t$, at time $t$. Let $\tilde{y}_t = \{\tilde{y}_1, \ldots, \tilde{y}_t\}$, where $|y_t - \tilde{y}_t| \leq \epsilon$. Suppose that $|\mu_t(x) - \tilde{\mu}_t(x)| < Ce^{\sqrt{x}}$ and $t$, where $\tilde{\mu}_t(x) = \kappa(x, X_t) (\kappa(X_t, X_t) + \eta I)^{-1} \tilde{y}_t$ and $C$ is a constant.

Theorem 7. Assume $g$ is constructed by sampling from $q \sim \mathcal{GP}(0, \kappa(\cdot))$, where $\kappa$ is a SE kernel. Define $\beta_t = O(d \log t)$. Running GP-UCB with $\beta_t$ and the biased samples $(x_i, \tilde{y}_t)$ instead of noisy samples $(x_i, y_t)$,
where $y_i = f(x_i) + \epsilon_i$ and $y_i = g(x_i) + \epsilon_i$. Assuming $|f(x) - g(x)| \leq \epsilon$, where $\epsilon$ is a constant, we then obtain the simple regret bound $O(\sqrt{\frac{\epsilon D}{T}} + \epsilon)$ with high probability.

**Corollary 8.** The simple regret of GP-UCB on function $f$ can be bounded as $O(\sqrt{\frac{\epsilon D}{T}} + \epsilon)$.

Theorem 7 and Corollary 8 are intuitive by considering the case that the bias is exactly $\epsilon$. If the approximation error is small, Add-GP-UCB and PP-GP-UCB can provide a satisfactory performance in practice, since the imperfect optimization of $\varphi_l$ in GP-UCB may cause larger error than $\epsilon$.

### 6.1 Validity of RPP-GP-UCB

We first consider the case when $\alpha = 0$, which implies $W$ satisfies the $\delta$-ratio constraint. Assume for simplicity that the optimal point $z_*$ is not on the boundary of the domain. Then the Hessian matrix $H_*$ of the $f$ at $z_*$ is negative definite. If $H_*$ further has small enough off-diagonal elements such that there exists a matrix $Q$ satisfying $\delta$-ratio constraint and $H_* = -QQ^T$. In turn, the following corollary holds.

**Proposition 9.** If $\tilde{f}$ satisfies Assumption 3, and the Hessian matrix of $f(z_*)$ can be decomposed as $-QQ^T$, where $Q$ satisfies the $\delta$-ratio constraint, there exists a projected-additive approximation $h$ such that $|f(z) - h(z)| = O(e^3)$, where $e = \|z - z_*\|$, and $z_*$ is the optimal point of $f$.

The mild-assumption in Corollary 9 is valid when the correlation between different dimensions is small. A special case is the additive function, where the correlation between different dimensions is zero.

If $\alpha > 0$, which means $W$ does not satisfy the $\delta$-ratio constraint, which is possible when $\delta$ is small. From the approximation perspectives as shown in Proposition 4 and Proposition 5, optimizing (8) is seeking for an intermediate approximation between the first-order and the second-order approximation. The result could be justified by Section 5.1.1 empirically. Even if we use small $\delta$, we could still get performance gain.

### 6.2 Practical Concern of $\delta$

If we set $\delta$ to be large, then we still need to search in a large domain. For example, the worst case is $\alpha = 1$ if $\delta$ is large enough. If we do not have enough computational budget to optimize the acquisition function, then we can only use small $\delta$ in practice. Note that $\delta$ determines how expressive our model is and therefore characterizes a bias variance tradeoff. If we set $\delta$ smaller, we increase the bias of the estimated $W$. However, note that the optimization of finding $W$ (Gilboa et al., 2013) is non-convex, and we are only able to find the local optimum. When $D$ is large, it is impossible to reconstruct $W \in \mathbb{R}^{D \times D}$ using a limited number of queries without any assumptions. Therefore, (7) plays a role as regularization to reduce the variance of estimating the transform matrix $W$ given $\delta$.

Intuitively, the RPP-GP-UCB is inspired from the geometrical interpretation to reduce the search space; algorithmically, RPP-GP-UCB seeks for a intermediates between the first-order and second-order approximation and balanced the trade-off between the increased search domain and the space of the function representation; empirically, we demonstrate RPP-GP-UCB outperforms existing approaches even though we use small $\delta$.

### 7 Conclusion

In this paper, we generalize the existing additive model to projected additive model with regret analysis. We propose RPP-GP-UCB to solve the difficulty of the increased function domain, and demonstrate the efficacy and validity of the proposed RPP-GP-UCB from both theoretical and empirical sides. We provide a theoretical study to justify the validity of the additive and projected-additive model, which could be treated as first-order and second-order approximation, respectively. Even the function is not projected-additive, the difference between achieved suboptimal and global optimal is bounded under certain conditions.

The empirical study on synthetic and real-world data demonstrates the proposed RPP-GP-UCB outperforms the additive model (Kandasamy et al., 2015) and low-dimensional assumption model (Wang et al., 2013). In practice, we observe that even if the small $\delta$, which only model partial correlation between different dimensions, could benefit a lot. On the other hand, if the low dimensional assumption holds, Wang et al. (2013) may have a promising result. However, it is possible that it still has dozens of reduced dimensions. As the empirical results suggest, RPP-GP-UCB could still benefit the reduced low dimensional problem.

Our work provides a more expressive framework for BO in high dimensions. Although we do not focus on perfect learning for RPP-GP, it already gives us a significant performance gain. However, we observe that the better optimization for RPP-GP results in better BO performance. Therefore, it worth studying the better learning algorithm for RPP-GP to further boost the performance.
References


