BOCK : Bayesian Optimization with Cylindrical Kernels Supplementary Materials

Anonymous Authors[1]

1. Special Treatment of the center point

In Section3.3, we propose the special treatment on the center point to correct the problem resulting from over-expansion of the center point. We provide a justification for the positive semi-definiteness of K_{cyl}.

Since the cylindrical kernel K_{cyl} is a tensor product of the kernel K_r from the radius component, and the kernel K_a from the angular component, if we can show that both K_r and K_d are proper kernels (*i.e.* positive semi-definite), then we can conclude that K_{cyl} is also a proper kernel (Rasmussen & Williams, 2006).

Let us denote with $T : B(\mathbf{0}, R) \rightarrow C(0, R; \mathbf{0}, 1)$ the transformation from a ball to a cylinder, and with π_a the projection to angle component in a cylinder. For a given set $\widetilde{\mathcal{D}} = \mathcal{D} \cup \{\mathbf{0}\}$, we denote the angle component $\pi_a(T(\mathcal{D}))$ as $\mathcal{D}_a$. Then the gram matrix of K_a on $\widetilde{\mathcal{D}}$ can be represented by

$$\begin{bmatrix} K_{cyl}(\mathcal{D}_a, \mathcal{D}_a) & K_a(\mathcal{D}_a, \mathbf{a}_{arbitrary}) \\ K_a(\mathbf{a}_{arbitrary}, \mathcal{D}_a) & K_a(\mathbf{a}_{arbitrary}, \mathbf{a}_{arbitrary}) \end{bmatrix} \tag{1}$$

In the special treatment, we set $\mathbf{a}_{arbitrary} = \mathbf{a}* = \mathbf{x}_* / \| \mathbf{x}_* \|$. This is nothing but the gram matrix of K_a on the dataset $\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_N, \mathbf{a}_*$ As long as, K_a is proper kernel, using the special treatment does break the positive semi-definiteness of kernel.

The special treatment assumes $\mathbf{x}_* \neq \mathbf{0}$. A single point is of measure zero under any non-atomic measure. This assumption can be safely made, theoretically. In our experiments, we start with data including $\mathbf{0}$ as an initial data point, thus the acquisition function does not need to go over $\mathbf{x}_* = 0$ anymore.

Interestingly, this special treatment bears similarity to Bayesian Optimization using treed Gaussian Processes (Assael et al., 2014). When there is $\mathbf{0}$ in our training data set, at each prediction, we have a Gaussian Process on the same data set but one point. Namely, one can view this as having different Gaussian Processes at different prediction points, in the sense that the data conditioning the Gaussian Process change (not the kernel parameters). As the treed Bayesian Optimization guarantees continuity between the regions having the different Gaussian Processes is also, the cylindrical kernel with the special treatment also has continuity since the Gram matrix is a continuous function of $\mathbf{b}_a rbitrary$.

However, at different prediction points we have different gram matrices. Hence, a naive implementation of the above idea makes the maximization of the acquisition function infeasible. In Gaussian process prediction, main computation bottle is to calculate a quadratic form as below

$$\begin{bmatrix} \mathbf{p}^T & p_0 \end{bmatrix} \left(\begin{bmatrix} K_{cyl}(\mathcal{D}, \mathcal{D}) & K_{cyl}(\mathcal{D}, 0) \\ K_{cyl}(0, \mathcal{D}) & K_{cyl}(0, 0) \end{bmatrix} + \sigma^2_{obs} I \right)^{-1} \begin{bmatrix} \mathbf{q}^T \\ q_0 \end{bmatrix} \tag{2}$$

Fortunately, we can calculate the quadratic form eq (2) efficiently by using block matrix inversion. Once we calculate $K_{cyl}(\mathcal{D}, \mathcal{D})^{-1}$, by using pre-calculated, $K_{cyl}(\mathcal{D}, \mathcal{D})^{-1}$, calculating eq (2) for different $\mathbf{x}_*$ requires marginal computation.

[1] Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

2. Implementation Detail

Parts of implementation details is provided in Section 4 except for the prior distribution we use for radius kernel warping eq 8. In order to make BOCK focus more on the center, we make prior concave and non-decreasing by using spike and slab prior (Ishwaran et al., 2005). In eq 8, $\log(\alpha)$ has spike and slab prior on positive real line. $\log(\beta)$ has spike and slab prior on negative real line.

3. Benchmark functions

The suggested search space for below benchmark functions are adjusted to be $[-1, 1]^D$ in our experiments.

3.1. Repeated Branin

$$f_{rep-branin}(x_1, x_2, \cdots, x_D) = 1/\lfloor\frac{D}{2}\rfloor \sum_{i=1}^{\lfloor D/2 \rfloor} f_{branin}(x_{2i-1}, x_{2i}) \tag{3}$$

where f_{branin} is branin function whose formula can be found in (Laguna & Martí, 2005). The original search space of branin function is $[-5, 10] \times [0, 15]$

3.2. Repeated Hartmann6

$$f_{rep-hartmann6}(x_1, x_2, \cdots, x_D) = 1/\lfloor\frac{D}{6}\rfloor \sum_{i=1}^{\lfloor D/6 \rfloor} f_{hartmann6}(x_{6i-5}, x_{6i-4}, x_{6i-3}, x_{6i-2}, x_{6i-1}, x_{6i}) \tag{4}$$

where $f_{hartmann6}$ is hartmann6 function whose formula can be found in (Laguna & Martí, 2005). The original search space of hartmann6 function is $[0, 1]^6$

3.3. Rosenbrock (Laguna & Martí, 2005)

$$f_{rosenbrock}(x_1, x_2, \cdots, x_D) = \sum_{i=1}^{D-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right] \tag{5}$$

The original search space is $[-5, 10]^D$

3.4. Levy (Laguna & Martí, 2005)

$$f_{levy}(x_1, x_2, \cdots, x_D) = \sin^2(\pi w_1) \sum_{i=1}^{D-1} (w_i - 1)^2 \big[1 + 100 \sin^2(\pi w_i + 1)\big] + (w_D - 1)^2 \big[1 + \sin^2(2\pi w_D) \tag{6}$$

$$w_i = 1 + \frac{x_i - 1}{4}$$

The original search space is $[-10, 10]^D$

4. Efficiency vs accuracy

We conduct the same analysis for efficiency vs accuracy with other benchmark functions on 20 dimensional case. In all cases, BOCK is the closest to the optimum operating point (0, 0) 1. Matern is also accurate enough, although considerably slower, while SMAC and additive BO are faster but considerably less accurate.

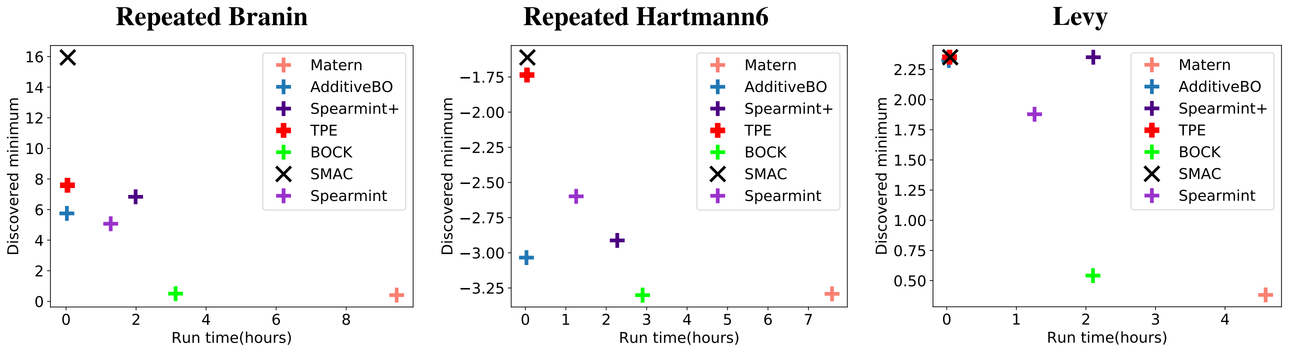

Figure 1. Accuracy of Bayesian Optimization methods vs wall clock time efficiency for the 20-dimensional Repeated Branin, Repeated Hartmann6, Levy benchmark. BOCK is the closest to the optimum operating point $(0, 0)$. Matern is also accurate enough, although considerably slower, while SMAC and additive BO are faster but considerably less accurate.

5. Scalability

We also conduct the experiment to check the scalability of algorithms with other benchmark functions on 20 and 100 dim. The same observation that BOCK is clearly more efficient and less effected by the increasing dimensionality can be made 2.

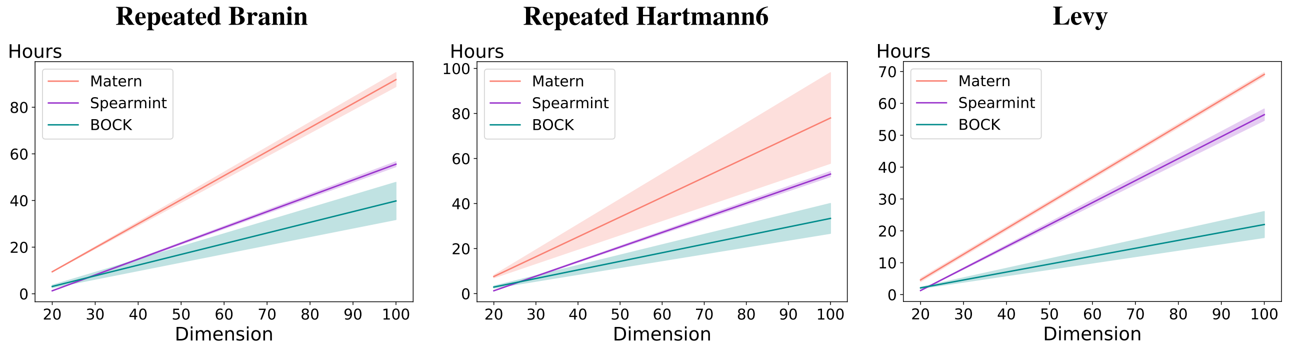

Figure 2. Wall clock time(hours) on the Repeated Branin, Repeated Hartmann6, Levy benchmark for an increasing the number of dimensions (20 and 100 dimensions, using 200 and 600 function evaluations respectively for all methods). The solid lines and colored regions represent the mean wall clock time and one standard deviation over these 5 runs. As obtaining the evaluation score $y = f(\mathbf{x}_*)$ on these benchmark functions is instantaneous, the wall clock time is directly related to the computational efficiency of algorithms. In this figure, we compare BOCK and BOs with relative high accuracy in all benchmark functions, such as Spearmint and Matern. BOCK is clearly more efficient, all the while being less affected by the increasing number of dimensions.

References

Assael, John-Alexander M, Wang, Ziyu, Shahriari, Bobak, and de Freitas, Nando. Heteroscedastic treed bayesian optimisation. *arXiv preprint arXiv:1410.7172*, 2014.

Ishwaran, Hemant, Rao, J Sunil, et al. Spike and slab variable selection: frequentist and bayesian strategies. *The Annals of Statistics*, 33(2):730–773, 2005.

Laguna, Manuel and Martí, Rafael. Experimental testing of advanced scatter search designs for global optimization of multimodal functions. *Journal of Global Optimization*, 33(2):235–255, 2005.

Rasmussen, Carl Edward and Williams, Christopher KI. *Gaussian processes for machine learning*, volume 1. MIT press Cambridge, 2006.