Are Random Decompositions all we need in High-Dimensional Bayesian Optimisation?

Juliusz Ziomek[*], Haitham Bou-Ammar[*†]

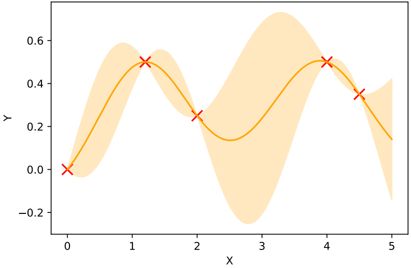
*Huawei Noah's Ark Lab, †University College London

ICML23, 23 – 29 Jul 2023

HUAWEI

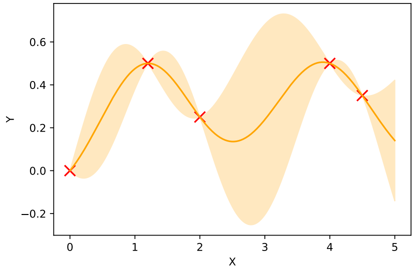
Bayesian Optimisation

- Bayesian Optimisation (BO) [Srinivas et al, 2010] aims to optimise a black-box function by using a surrogate Gaussian Process (GP) model



Bayesian Optimisation

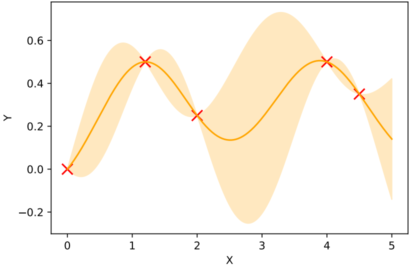
- Bayesian Optimisation (BO) [Srinivas et al, 2010] aims to optimise a black-box function by using a surrogate Gaussian Process (GP) model



- Based on the GP model, we optimise **acqusition function** to find most promising point to query next

Bayesian Optimisation

- Bayesian Optimisation (BO) [Srinivas et al, 2010] aims to optimise a black-box function by using a surrogate Gaussian Process (GP) model



- Based on the GP model, we optimise **acqusition function** to find most promising point to query next
- This works great with a small number of dimensions; struggles in high-dimensional spaces

Bayesian Optimisation with Additive Functions

- One solution: Assume the additive function ([Kandasamy et al, 2015], [Rolland et al, 2018], [Han et al, 2021])

$$f(\boldsymbol{x}) = \sum_{c \in g} f_c(\boldsymbol{x}_{[c]})$$

for each group of dimensions c in decomposition g, for example if $g = \{(1,4),(2),(3)\}$:

$$f(x_1, x_2, x_3, x_4) = f_{1,4}(x_1, x_4) + f_2(x_2) + f_3(x_3)$$

HUAWEI

Bayesian Optimisation with Additive Functions

- One solution: Assume the additive function ([Kandasamy et al, 2015], [Rolland et al, 2018], [Han et al, 2021])

$$f(\boldsymbol{x}) = \sum_{c \in g} f_c(\boldsymbol{x}_{[c]})$$

for each group of dimensions c in decomposition g, for example if $g = \{(1,4),(2),(3)\}$:

$$f(x_1, x_2, x_3, x_4) = f_{1,4}(x_1, x_4) + f_2(x_2) + f_3(x_3)$$

- Problem: If the function is black-box, we do not know g

HUAWEI

Bayesian Optimisation with Additive Functions

- One solution: Assume the additive function ([Kandasamy et al, 2015], [Rolland et al, 2018], [Han et al, 2021])

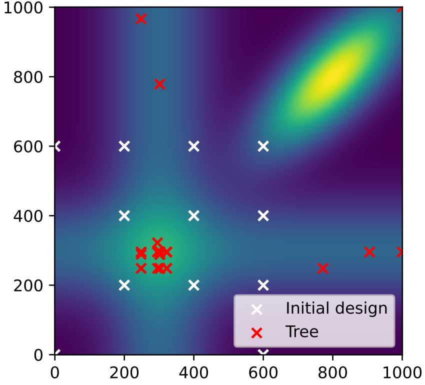
$$f(\boldsymbol{x}) = \sum_{c \in g} f_c(\boldsymbol{x}_{[c]})$$

  for each group of dimensions c in decomposition g, for example if $g = \{(1,4),(2),(3)\}$:

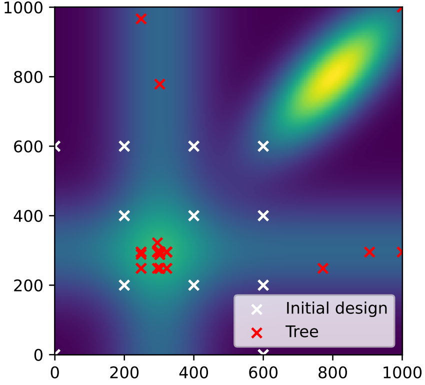
$$f(x_1, x_2, x_3, x_4) = f_{1,4}(x_1, x_4) + f_2(x_2) + f_3(x_3)$$

- Problem: If the function is black-box, we do not know g
- Existing methods learn g by maximum likelihood by selecting g that produces model with highest marginal likelihood $p(\mathcal{D}|g)$

Misleading decomposition learners

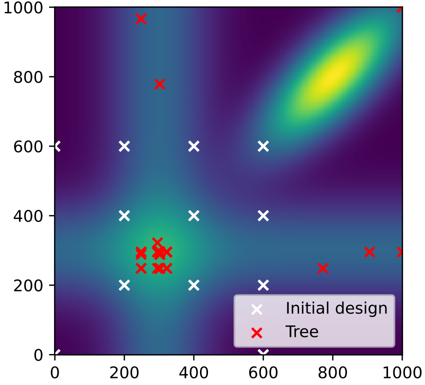


Misleading decomposition learners



- State-of-art Tree algorithm [Han et al, 2021] gets stuck in a sub-optimal mode

Misleading decomposition learners



- State-of-art Tree algorithm [Han et al, 2021] gets stuck in a sub-optimal mode
- This is because, in BO we have limited, local data
  $\rightarrow$ hard to extrapolate, easy to overfit

Analysing decomposition rules

- Instead of relying on limited, local data, let us consider data-independent pre-defined schemes for choosing decompositions

Theorem (Corollary 4.2 in the paper)

Let the black-box function f be selected by an adversary from an RKHS $\mathcal{H}^g$ of kernel k^g, defined over some decomposition g that is also selected by an adversary. After T rounds, a UCB-style BO algorithm with an $S(t) : \mathbb{Z}^+ \to \mathcal{G}$ decomposition rule, incurs with a probability of at least $1 - \delta_a - \delta_B$ the following total cumulative regret R_T:

$$R_T = \mathcal{O}\left(\sqrt{T \underbrace{\gamma_T}_{\text{Kernel Complexity}}}\left(B + \sqrt{\ln \frac{1}{\delta_A} + \underbrace{\gamma_T}_{\text{Kernel Complexity}}} + \frac{1}{\delta_B} \underbrace{\mathbb{E}_S\left[\sum_{t=1}^{T} \epsilon_t\right]}_{\text{Expected mismatch}}\right)\right),$$

where $B = \max_{t \in T} \|\hat{f}_t\|_t$ and $\|\cdot\|_t$ denotes the norm in $\mathcal{H}_t$.

Analysing decomposition rules

- Want both γ_t and ϵ_t to be "small"

Analysing decomposition rules

- Want both γ_t and ϵ_t to be "small"
- Making γ_t "small"

  γ_t measures the "complexity" of proposed decompositions. If we restrict our scheme to only suggest **tree decompositions** [Han et al, 2021], we can favourably bound γ_t

HUAWEI

Analysing decomposition rules

- Want both γ_t and ϵ_t to be "small"
- Making γ_t "small"
  γ_t measures the "complexity" of proposed decompositions. If we restrict our scheme to only suggest **tree decompositions** [Han et al, 2021], we can favourably bound γ_t
- Making ϵ_t "small"
  ϵ_t measures the mismatch between the true decomposition and the one we proposed. We show that for tree decomp., the scheme with lowest mismatch **chooses decompositions uniformly at random**.

HUAWEI

Analysing decomposition rules

- Want both γ_t and ϵ_t to be "small"
- Making γ_t "small"
  γ_t measures the "complexity" of proposed decompositions. If we restrict our scheme to only suggest **tree decompositions** [Han et al, 2021], we can favourably bound γ_t
- Making ϵ_t "small"
  ϵ_t measures the mismatch between the true decomposition and the one we proposed. We show that for tree decomp., the scheme with lowest mismatch **chooses decompositions uniformly at random**.

- **Our algorithm should select tree decompositions randomly!**

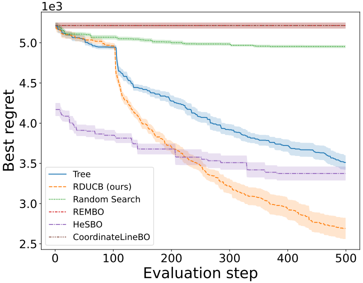
HUAWEI

Practical Algorithm

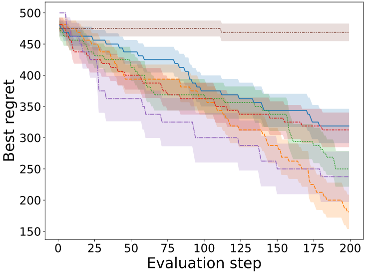
Algorithm RDUCB

Inputs: Black-box function f, evaluation budget N, initial budget N_{init}, exploration bonuses $\{\beta_t\}_{t=1}^{N}$
Evaluate N_{init} random inputs in f & populate $\mathcal{D}_{N_{\text{init}}}$
for $t = N_{\text{init}} + 1$ **to** N **do**
    Sample tree decomposition g
    Fit a GP using $\mathcal{D}_{t-1}$ with the kernel $k_g(\cdot)$
    Maximise $\alpha_t^{(\text{add-UCB})}(\boldsymbol{x}|\mathcal{D}_{t-1})$ with message passing
    Evaluate f on the suggested query & add to $\mathcal{D}_{t-1}$
end for

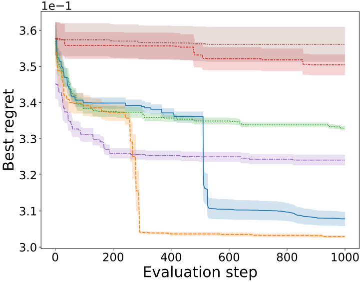
Selected Empirical Results



(a) 250-d Stybtang Function



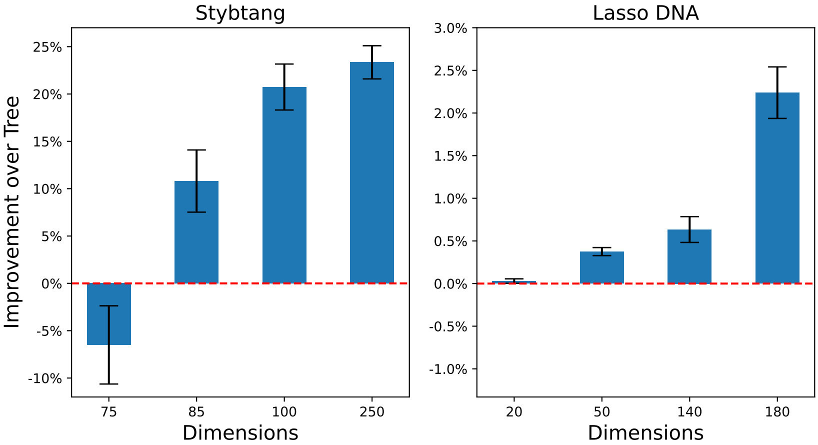
(b) 74-d `misc05inf` MIP Task



(c) 180-d DNA LassoBench Dataset

HUAWEI

Performance as dimensionality increases



Plug&Play for HEBO [Cowen-Rivers et al, 2022]

HEBO = Multi-acquisition + Input warping + Evolution + BO

RDHEBO = Random Decompositions + HEBO

Problem	**HEBO**	**RDHEBO**
MLP-Adam	92.68 ± 0.22	$\mathbf{93.67 \pm 0.30}$
MLP-SGD	90.66 ± 0.81	$\mathbf{91.65 \pm 0.10}$
DT	79.42 ± 0.45	$\mathbf{80.79 \pm 0.15}$
RF	84.97 ± 0.32	$\mathbf{87.64 \pm 2.00}$
Average	86.93 ± 0.45	$\mathbf{88.44 \pm 0.64}$

HUAWEI

References

Srinivas et al (2010)
Gaussian process optimization in the bandit setting: No regret and experimental design, ICML

Kandasamy et al (2015)
High dimensional Bayesian optimisation and bandits via additive models, ICML

Rolland et al (2018)
High-dimensional Bayesian optimization via additive models with overlapping groups, AISTATS

Han et al (2021)
High-dimensional Bayesian optimization via tree-structured additive models, AISTATS

Cowen-Rivers et al (2022)
HEBO: pushing the limits of sample-efficient hyper-parameter optimisation, JAIR