High dimensional Bayesian optimization

Zhitong

Layout

- Introduction of common approaches to high-dimensional(D > 15) Bayesian optimization.
- Paper 1: Randomly Projected Additive Gaussian Processes for Regression [Delbridge et al. ICML 2020]
- Some background for learning decomposition with additive models.
- Paper 2: Are Random Decompositions all we need in High Dimensional Bayesian Optimisation? [Ziomek et al. ICML 2023]
- Discussion

Common approaches(assumptions) to HDBO

- Low dimensional structure(Linear projection), e.g., SaasBO, ALEBO, REMBO.
  - Main idea: Assumes $f(x) \approx g(\phi x)$ for d << D.
  - Failure mode: f(x) may not have low-dimensional structure, sensitive to choice of d. RemBo always fails empirically, poor GP fit.

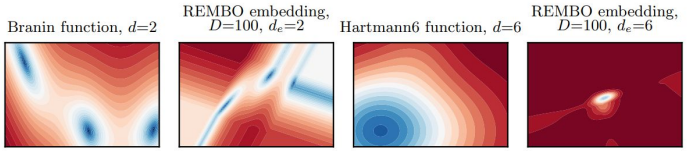


Figure 1: A visualization of REMBO embeddings for two test functions. *(Far left)* The Branin function, d=2, extended to D=100. *(Center left)* A REMBO embedding of the D=100 Branin function. *(Center right)* A center slice of the d=6 Hartmann6 function, similarly extended to D=100. *(Far right)* The same slice of a REMBO embedding of that function. The embedding produces distortions and non-stationarity in the function that render it difficult to model.

Common approaches(assumptions) to HDBO

- Structure decomposition(Additive, Add-GP-UCB)
  - Main idea: Assumes high-dimensional function decomposes into a sum of low-dim functions.
  - Failure mode: Additive structure is usually not a realistic assumption. Learning the decomposition is hard.
- Local optimization (TuRBO)
  - Main idea: Restrict where the acquisition function is optimized to avoid over-exploration.
  - Failure mode: TuRBO converge slowly as it is designed for high-throughput setting.

Additive structures in GP

- Generalized additive model(GAM): $k(x, x') = \sum k_j(x(j), x'(j))$ f(x1,x2,x3) = f'(x1)+f''(x2)+f'''(x3)
  - Issue: Model assumption is too strong.
- Learned decomposition: $k(x, x') = \sum k_j(x^{(j)}, x'^{(j)})$ f(x1,x2,x3) = f'(x1,x2) + f''(x3)
  - Issue: Decomposition is hard to learn.
- Projection Pursuit: $k(x, x') = \sum k_j(\eta_j^T x, \eta_j^T x')$
  - Issue: Learning projection by sampling is slow and hard. Learning projection by gradient method will overfit.
- Random projection: $k(x, x') = \sum k_j(P_j x, P_j x')$
  - Issue: Often needs $J \approx d$ number of random embeddings for d < 500, sensitive to the choice of J.

Method

$$k_{rp}(\boldsymbol{x}, \boldsymbol{x}') = \sum_{j=1}^{J} \alpha_j k_j(P^{(j)}\boldsymbol{x}, P^{(j)}\boldsymbol{x}'), \tag{2}$$

$$\forall j \in [J], \quad P^{(j)} \in \mathbb{R}^{D_j \times d}, \tag{3}$$

$$P^{(j)}_{r,c} \sim \mathcal{N}\left(0, \frac{1}{D_j}\right) \quad \forall r \in [D_j], c \in [d]. \tag{4}$$

Propositions and proofs(1D projection)

Proposition 1. *Let $\phi\colon \mathbb{R} \mapsto [-1,1]$ be a 1-dimensional kernel, and let $(\boldsymbol{\eta}_j : j \geq 1)$ be an i.i.d. sequence of random variables in $\mathbb{R}^d$ drawn from a common isotropic distribution $\mathcal{D}$. Then, for some expected kernel $k_{expected}\colon \mathbb{R} \mapsto [0,1]$, for any $\boldsymbol{\tau} \in \mathbb{R}^d$, almost surely*

$$\lim_{J\to\infty} \frac{1}{J} \sum_{j=1}^{J} \phi(\boldsymbol{\eta}_j^\top \boldsymbol{\tau}) = \mathbb{E}[\phi(\eta_{11}||\boldsymbol{\tau}||_2)] =: k_{expected}(||\boldsymbol{\tau}||_2).$$

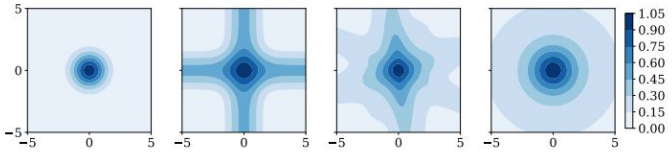


Figure 1. Contour plots of 2-dimensional kernels. From left to right: RBF, GAM RBF, RPA-GP with 16 projections, and DPA-GP with 16 projections. With enough additive projections, we attain approximately spherical covariance, and choosing well-placed directions facilitates convergence.

Propositions and proofs(1D projection)

Corollary 1. *If* $\phi(x) = e^{-\frac{1}{2}x^2}$ *and* $\boldsymbol{\eta}_1 \sim \mathcal{N}(0, I_d)$, *then*

$$k_{expected}(\boldsymbol{\tau}) = \frac{1}{\sqrt{1 + ||\boldsymbol{\tau}||_2^2}} \triangleq k_{IMQ}(\boldsymbol{\tau}). \tag{5}$$

Corollary 2. *If* $\phi(x) = \cos(x)$ *and* $\boldsymbol{\eta}_1 \sim \mathcal{N}(0, I_d)$, *then*

$$k_{expected}(\boldsymbol{\tau}) = e^{-\frac{1}{2}||\boldsymbol{\tau}||_2^2} \triangleq k_{RBF}(\boldsymbol{\tau}). \tag{6}$$

Trick 1(max separation)

Goal: Decrease the number of random projections needed for convergence to limiting(expected) kernel.

Proposition 2. *Let ϕ, $k_{expected}$ be as in Proposition 1. Let $\{\boldsymbol{\eta}_j\}_{j=1}^J$ be a sequence of random variables drawn i.i.d. from an isotropic distribution. Let $\delta > 0$. Then, with probability at least $1-\delta$, we have simultaneously for all pairs of points $\boldsymbol{\tau}_{i,k}$, $i,k \in [n]$,*

$$\left|\frac{1}{J}\sum_{j=1}^{J}\phi(\boldsymbol{\eta}_j^\top \boldsymbol{\tau}_{i,k}) - k_{expected}(||\boldsymbol{\tau}_{i,k}||_2)\right| \leq \frac{2}{3J}(\log(1/\delta)+2\log(n)+1) + \sqrt{\frac{2\sup_{i,k}\mathrm{var}(\phi(\boldsymbol{\eta}_1^\top \boldsymbol{\tau}_{i,k}))}{J}}$$

Definition of separation:

$$\delta(\boldsymbol{\eta}_1,...,\boldsymbol{\eta}_J) = \min_{j\neq j'} \cos^{-1}(|\boldsymbol{\eta}_j^\top \boldsymbol{\eta}_{j'}|),$$

Objective to minimize (pow of 4):

$$\ell(\boldsymbol{\eta}_1,...,\boldsymbol{\eta}_J) = \sum_{j\neq j'}(\boldsymbol{\eta}_j^\top \boldsymbol{\eta}_{j'})^4.$$

Trick 2 (Adding ARD)

Question: Should we add ARD before projection or after projection?

$$k_{rpARD}(\boldsymbol{x}, \boldsymbol{x}') = \sum_{j=1}^{J} \alpha_j k_j(P^{(j)} A\boldsymbol{x}, P^{(j)} A\boldsymbol{x}'),$$

The main idea of the first paper is that: Learning true decomposition is hard, learning projection will overfit, let’s use random projection instead.

Experiment results of Paper 1

Low dimensions convergence:

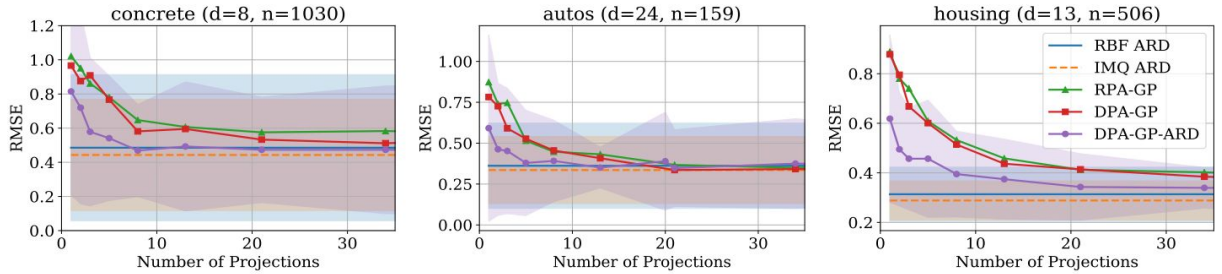
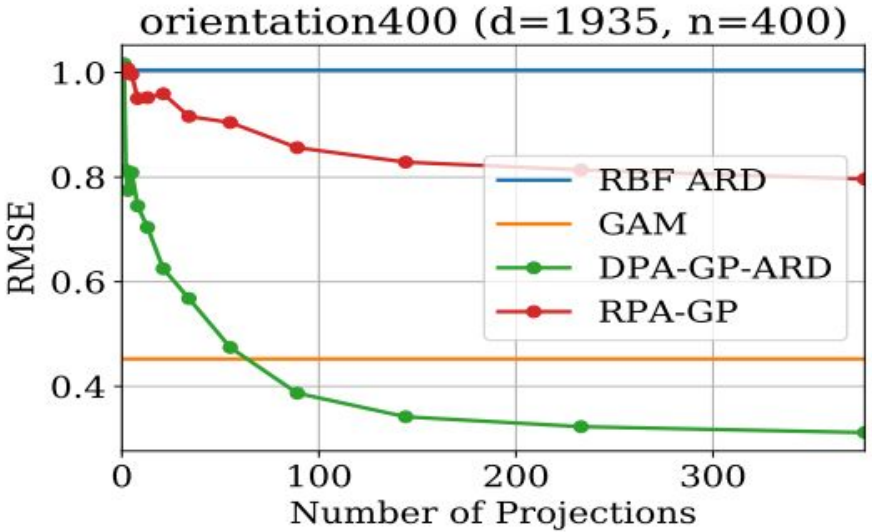


Figure 3. Representative test RMSE of RPA-GP and DPA-GP as the number of projections vary compared to full-dimensional RBF and inverse multiquadratic (IMQ) kernels. Shaded regions are 2 times the standard deviation over cross-validation, and lines are the average RMSE. For clarity, we only show the variation for DPA-GP-ARD. In general, there is a fast convergence to the performance of RBF and IMQ kernels, and DPA-GP consistently improves upon RPA-GP by a small amount, and applying length-scales before (DPA-GP-ARD) projection dramatically increases performance.

Experiment results of Paper 1

Performance on high dimensions:



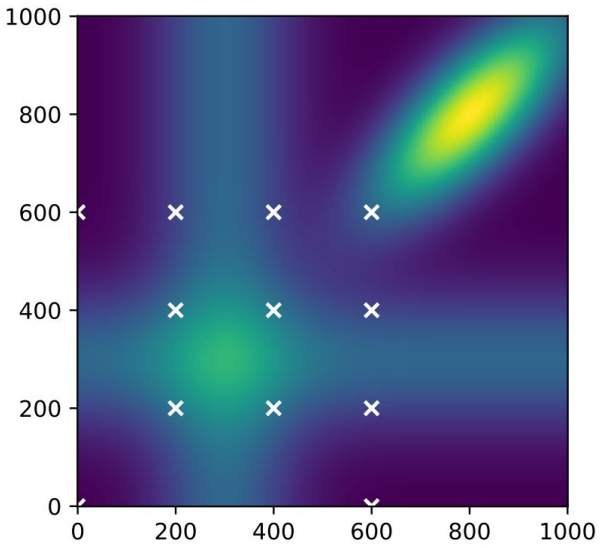
Background on additive model with decomposition

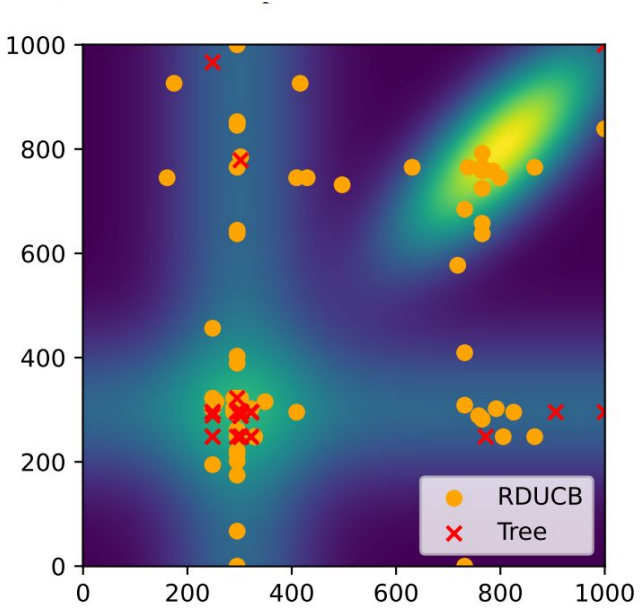
- **High Dimensional Bayesian Optimisation and Bandits via Additive Models** [Kandasamy et al. ICML 2015], proposed additive models for BO. Learn decomposition by maximizing the GP marginal likelihood. (randomly select O(D) decompositions out of D!*M!/d!^M for every N steps in BO loop.)
- **High-Dimensional Bayesian Optimization via Additive Models with Overlapping Groups**[Rolland et al. AISTATS 2018], assumes decomposition graph to be a potential fully connected graph. Place prior on edges, and use Gibbs sampling to learn dependency graph. (D(D-1)/2 parameters on graph.)
- **High-Dimensional Bayesian Optimization via Tree-Structured Additive Models** [Han et al. 2021 AAAI] Assumes decomposition to be strict tree structure. Easier for message passing.

What’s the problem now?

Recall that we learn decomposition from data, what if training data is “misleading”? Especially for the BO setting, where data is limited.

Example in RDUCB



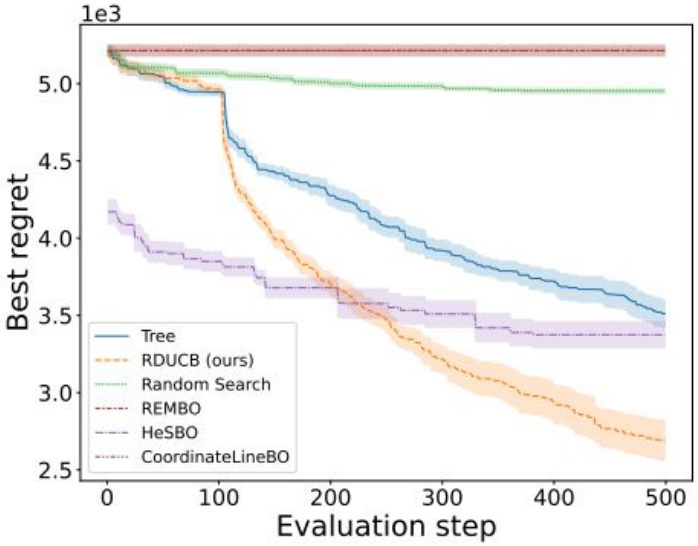


RDUCB algorithm

Algorithm 1 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 (Alg. 2)
    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

Empirical result



(a) 250-d Stybtang Function

Reference:

Papers:

Are Random Decompositions all we need in High Dimensional Bayesian Optimisation? [Ziomek et al.]

Randomly Projected Additive Gaussian Processes for Regression[Delbridge et al.]

High Dimensional Bayesian Optimisation and Bandits via Additive Models [Kandasamy et al.]

High-Dimensional Bayesian Optimization via Additive Models with Overlapping Groups [Rolland et al]

High-Dimensional Bayesian Optimization via Tree-Structured Additive Models [Han et al.]

High-Dimensional Bayesian Optimization with Sparse Axis-Aligned Subspaces [Eriksson et al.]

Re-Examining Linear Embeddings for High-Dimensional Bayesian Optimization [Letham et al.]

Bayesian Optimization in a Billion Dimensions via Random Embeddings [Wang et al]

Reference:

Talks:

https://slideslive.com/38928196

https://slideslive.com/39002628

https://slideslive.com/38937117

(619) David Eriksson | "High-Dimensional Bayesian Optimization" - YouTube