Hyperparameter Transfer Learning with Adaptive Complexity

Samuel Horváth[1], Aaron Klein[2], Peter Richtárik[1] and Cédric Archambeau[2]

King Abdullah University of Science and Technology[1]
Amazon Web Services[2]

Problem setup

We consider a standard **multi-task Bayesian Optimization (BO)** framework, where one aims to optimize an expensive black-box function $f_{T+1} : \mathcal{X} \to \mathbb{R}$ with a minimal number of function evaluations:

$$\mathbf{x}^\star_{T+1} = \text{argmin}_{\mathbf{x}\in\mathcal{X}} f_{T+1}(\mathbf{x}), \tag{1}$$

where $\mathcal{X} \subset \mathbb{R}^D$ denotes the configuration space.
We assume that we have already **completed T related Hyperparameter Optimization (HPO) tasks** $\{f_1, \dots f_T\}$ that share the same configuration space $\mathcal{X}$ and we have access to

$$\mathcal{D} = \left\{D_t : D_t = \{(\mathbf{x}_{tn}, y_{tn})\}_{n=1}^{N_t}\right\}_{t=1}^{T},$$

the data collected while optimizing the set of black-box functions $\{f_t\}_{t=1}^T$, $y_{tn} = f_t(\mathbf{x}_{tn})$.

Contributions

We resolve a deficiency of popular Adaptive Bayesian Linear Regression (ABLR) [1] applied to multi-task BO, which is that **the number of non-linear basis functions is not adapted to the target task**, where the number of observations is typically smaller than in the previous tasks, making it prone to overfitting. This issue is due to the direct use of conventional multi-task models in the context of sequential decision making problems such as BO. To solve this issue:

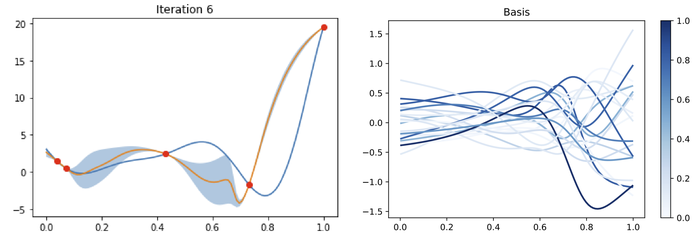
- We use **nested dropout** [2] to learn an **ordered set of features for transfer learning** in the context of BO. To the best of our knowledge, no other multi-task learning approach proposed in the literature is able to learn features that take the adaptive complexity of the target task into account.
- We use **Automatic Relevance Determination (ARD)** to automatically determine which basis functions to activate at transfer in a data-driven fashion. Hence, the resulting transfer learning model is able to adapt its capacity to the amount of data available in the target task.
- We show that we can improve the sample efficiency of multi-task BO and **avoid overfitting** in low data regimes without hurting the transfer learning performance in high data regimes.

ABLR with Adaptive Complexity (ABRAC)

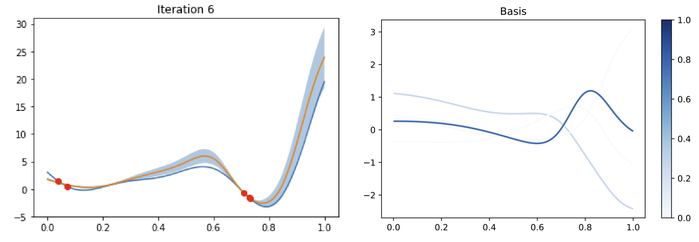
Algorithm 1 ABRAC

Input: number of initial points n_0, budget N, feature net $\phi_z(\cdot) : \mathbb{R}^P \to \mathbb{R}^d$ parametrized by z, filter F^k, previous evaluations $\{\{(x_{ti}, y_{ti})\}_{i=1}^{N_t}\}_{t=1}^{T}$.
Fit $\phi_{\mathbf{z}}(\cdot)$ using $\{\{(x_{ti}, y_{ti})\}_{i=1}^{N_t}\}_{t=1}^{T}$ using random truncation (nested dropout) in the final layer.
Observe f_{T+1} at n_0 randomly selected points $x_1, x_2, \dots, x_{n_0} \in \mathcal{X}$.
$\mathcal{C} = \{(x_i, y_i)\}_{i=1}^{n_0}$, where $y_i = f_{T+1}(x_i)$.
Set $n = n_0$.
while $n < N$ **do**
  Fit probabilistic model g via ARD.
  $x_n = \text{argmax}_{x\in\mathcal{X}} A_g(x)$, where A is a given acquisition function.
  Observe $y_n = f_{T+1}(x_n)$.
  Update $\mathcal{C} \leftarrow \mathcal{C} \cup \{(x_n, y_n)\}$, $n \leftarrow n + 1$
end while
Output: $\hat{x} = \text{argmin}_{i=1,2,\dots,N} f_{T+1}(x_i)$

Fit Visualization on Forrester Functions

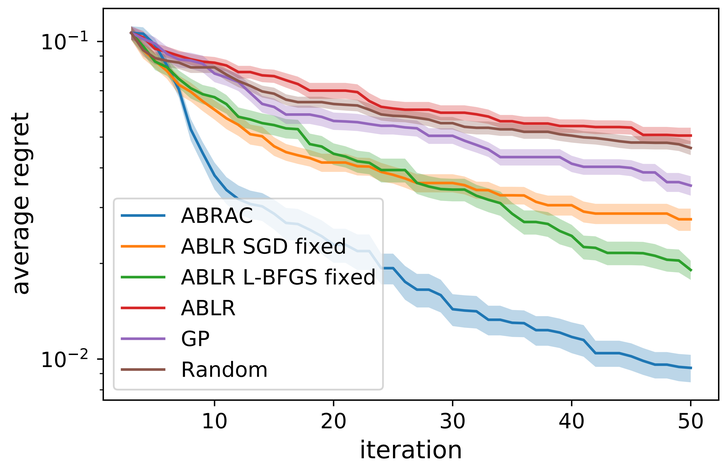


(a) `ABLR`



(b) `ABRAC`

Neural Architecture Search



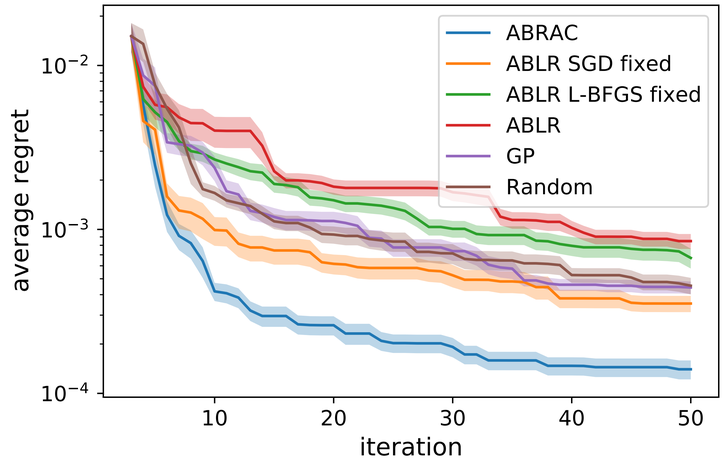


Figure: Tabular benchmarks by Klein and Hutter (2019). Top: Protein Structure, bottom: Slice Localization.

References

[1] Valerio Perrone, Rodolphe Jenatton, Matthias Seeger, and Cédric Archambeau. Scalable hyperparameter transfer learning. In *NeurIPS*, 2018.

[2] Oren Rippel, Michael Gelbart, and Ryan Adams. Learning ordered representations with nested dropout. In *ICML*, 2014.