An Efficient Approach for Assessing Parameter Importance in Bayesian Optimization

Frank Hutter
Freiburg University
fh@informatik.uni-freiburg.de

Holger Hoos and Kevin Leyton-Brown
University of British Columbia
{hoos, kevinlb}@cs.ubc.ca

UNI FREIBURG

UBC

… in 20 sec

- **Bayesian Optimization** is a powerful technique for finding the global optimizer of blackbox functions.
- Users want to know more: which inputs are important, the effects of which ones are correlated?
- We use functional ANOVA to provide such information, based on efficient operations in random forests.

Efficient Marginal Performance Predictions in Random Forests

Basic Definitions (only for reference)

Let A be an algorithm having n parameters with domains $\Theta_1, \ldots, \Theta_n$. We use integers to denote the parameters, and N to refer to the set $\{1, \ldots, n\}$ of all parameters of A.

Definition 1 (Configuration space $\boldsymbol{\Theta}$). *A's* configuration space *is* $\boldsymbol{\Theta} = \Theta_1 \times \cdots \times \Theta_n$.

Definition 2 (Parameter Instantiation). *A* complete instantiation *of an algorithm's n parameters is a vector $\boldsymbol{\theta} = \langle \theta_1, \ldots, \theta_n \rangle$ with $\theta_i \in \Theta_i$. We also refer to complete parameter instantiations as* parameter configurations. *A* partial instantiation *of a subset $U = \{u_1, \ldots, u_m\} \subseteq N$ of A's parameters is a vector $\boldsymbol{\theta}_U = \langle \theta_{u_1}, \ldots, \theta_{u_m} \rangle$ with $\theta_{u_i} \in \Theta_{u_i}$.*

Definition 3 (Extension Set). *Let $\boldsymbol{\theta}_U = \langle \theta_{u_1}, \ldots, \theta_{u_m} \rangle$ be a partial instantiation of the parameters $U = \{u_1, \ldots, u_m\} \subseteq N$. The* extension set $X(\boldsymbol{\theta}_U)$ of $\boldsymbol{\theta}_U$ *is then the set of parameter configurations $\boldsymbol{\theta}_{N|U} = \langle \theta'_1, \ldots, \theta'_n \rangle$ such that $\forall j (j = u_k \Rightarrow \theta'_j = \theta_{u_k})$.*

Definition 4 (Range size). *The* range size $||S||$ *of an empty set S is $||\emptyset|| = 1$; for other finite S, the range size equals the cardinality: $||S|| = |S|$; and for closed intervals $S = [l, u] \subset \mathbb{R}$, $||S|| = u - l$. For cross-products $S = S_1 \times \cdots \times S_k$, $||S|| = \prod_{i=1}^{k} ||S_i||$.*

Main Definition and Results

Definition 5 (Marginal performance). *Let A's (true) performance be $y : \boldsymbol{\Theta} \mapsto \mathbb{R}$, $U \subseteq N$, and $T = N \setminus U$. A's* marginal performance $a_U(\boldsymbol{\theta}_U)$ *is then defined as*

$$a_U(\boldsymbol{\theta}_U) = \mathbb{E}[y(\boldsymbol{\theta}_{N|U}) \mid \boldsymbol{\theta}_{N|U} \in X(\boldsymbol{\theta}_U)] = \frac{1}{||\boldsymbol{\Theta}_T||} \int y(\boldsymbol{\theta}_{N|U}) d\boldsymbol{\theta}_T.$$

Similarly, A's marginal predicted *performance $\hat{a}_U(\boldsymbol{\theta}_U)$ under a model $\hat{y} : \boldsymbol{\Theta} \to \mathbb{R}$ is*

$$\hat{a}_U(\boldsymbol{\theta}_U) = \frac{1}{||\boldsymbol{\Theta}_T||} \int \hat{y}(\boldsymbol{\theta}_{N|U}) d\boldsymbol{\theta}_T. \quad (1)$$

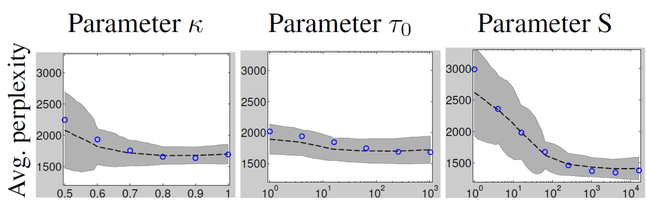
Theorem 6. *Given the partitioning $\mathcal{P}$ of a regression tree $\mathcal{T}$ that defines a predictor $\hat{y} : \boldsymbol{\Theta} \mapsto \mathbb{R}$, and a partial instantiation $\boldsymbol{\theta}_U$ of $\boldsymbol{\Theta}$'s parameters N, $\mathcal{T}$'s marginal prediction $\hat{a}_U(\boldsymbol{\theta}_U)$ can be computed as*

$$\hat{a}_U(\boldsymbol{\theta}_U) = \sum_{P_i \in \mathcal{P}} \frac{||\boldsymbol{\Theta}^{(i)}_{N \setminus U}||}{||\boldsymbol{\Theta}_{N \setminus U}||} \mathbb{I}(\boldsymbol{\theta}_U \in \boldsymbol{\Theta}^{(i)}_U) \cdot c_i.$$

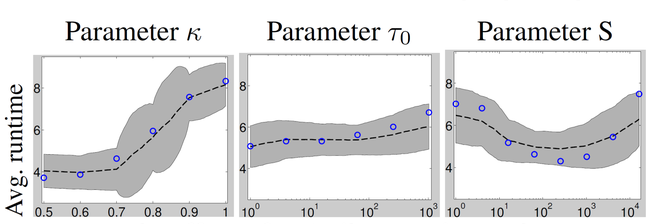
Corollary 8. *Given a random forest F with B trees of up to L leaves that defines a predictor $\hat{y} : \boldsymbol{\Theta} \to \mathbb{R}$ for a configuration space with n parameters and maximal categorical domain size D, the time and space complexity of computing a single marginal prediction of F is $O(B \cdot L \cdot \max\{D + n, n \log D\})$. Additional marginal predictions cost additional space $O(1)$ and time $O(B \cdot L \cdot n \log D)$.*

Experiment (on ground truth data)

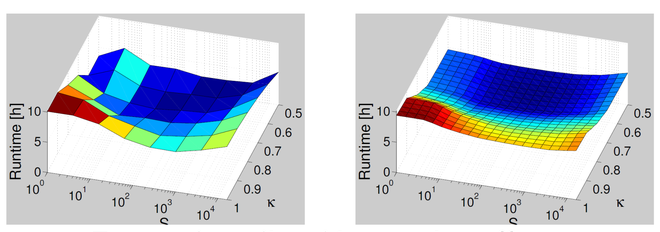
Online LDA [Hoffman et al, '10] with 3 parameters.
Performance known for 288-point grid;
100 randomly sampled grid points used for training



Main effects for Online LDA's perplexity



Main effects for Online LDA's runtime



True and predicted interaction effect

Efficient Decomposition of Variance

Functional ANOVA (not new)

Functional ANOVA decomposes a function $\hat{y} : \Theta_1 \times \cdots \times \Theta_n \to \mathbb{R}$ into additive components that only depend on subsets of its parameters N:

$$\hat{y}(\boldsymbol{\theta}) = \sum_{U \subseteq N} f_U(\boldsymbol{\theta}_U).$$

The components $f_U(\boldsymbol{\theta}_U)$ are defined as follows:

$$f_U(\boldsymbol{\theta}_U) = \begin{cases} \frac{1}{||\boldsymbol{\Theta}||} \int \hat{y}(\boldsymbol{\theta}) d\boldsymbol{\theta} & \text{if } U = \emptyset. \\ a_U(\boldsymbol{\theta}_U) - \sum_{W \subsetneq U} f_W(\boldsymbol{\theta}_W) & \text{otherwise.} \end{cases} \quad (4)$$

The constant $f_\emptyset$ is the function's mean across its domain. The unary functions $f_{\{j\}}(\boldsymbol{\theta}_j)$ are called *main effects* and capture the effect of varying parameter j, averaging across all instantiations of all other parameters.

By definition, the variance of $\hat{y}$ across its domain $\boldsymbol{\Theta}$ is

$$\mathbb{V} = \frac{1}{||\boldsymbol{\Theta}||} \int (\hat{y}(\boldsymbol{\theta}) - f_\emptyset)^2 d\boldsymbol{\theta}, \quad (5)$$

and functional ANOVA decomposes this variance into contributions by all subsets of variables (see, e.g., Hooker, 2007, for a derivation):

$$\mathbb{V} = \sum_{U \subset N} \mathbb{V}_U, \quad \text{where} \quad \mathbb{V}_U = \frac{1}{||\boldsymbol{\Theta}_U||} \int f_U(\boldsymbol{\theta}_U)^2 d\boldsymbol{\theta}_U.$$

The importance of all main and interaction effects f_U can thus be quantified by the fraction of variance they explain: $\mathbb{F}_U = \mathbb{V}_U / \mathbb{V}$.

Complexity with Random Forests

We can use our efficient marginal computations to compute these importance indices efficiently:

Theorem 9. *Given a configuration space $\boldsymbol{\Theta}$ consisting of n categorical[2] parameters of maximal domain size D and a regression tree $\mathcal{T}$ with L leaves that defines a predictor $\hat{y} : \boldsymbol{\Theta} \to \mathbb{R}$, we can exactly compute the fractions of variance explained by all subsets U of $\boldsymbol{\Theta}$'s parameters N of arity up to K, with space complexity $O(L \cdot D + L \cdot n)$ and time complexity $O\left(L \cdot D + \sum_{k=1}^{K} \binom{n}{k} \cdot D^k (L \cdot n \log d + 2^k)\right)$.*

To compute parameter importance in random forests, we simply apply Algorithm 2 for each tree, and compute means and standard deviations across the results.

How to Use This in Practice

- Collect performance data by running the algorithm with different parameter settings (e.g., run Bayesian Optimization)
- Fit a random forest model on that data (can e.g., be the model already used in BayesOpt)
- Determine important (pairs of) variables
- Inspect important main and interaction effects

- Future work: use within Bayesian optimization to iteratively focus on important parameters

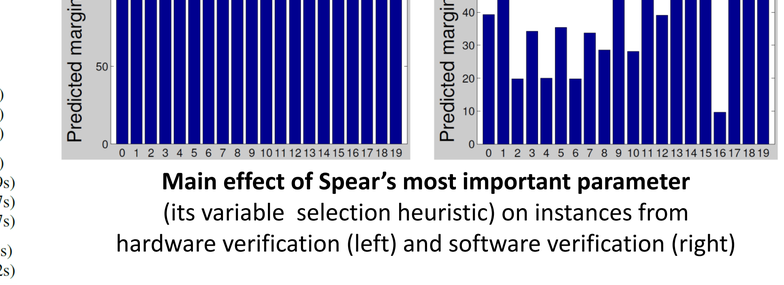
Application to Auto-WEKA [Thornton et al, 2013]

768 (!) parameters
Four parameters consistently turned out to be important:

- Machine learning algorithm (out of 31 choices)
- Base algorithm to use in an ensemble
- Feature selection: scoring mechanism for feature subsets
- Feature search: search mechanism through feature subsets

Main effect of the choice of machine learning algorithm

for CIFAR-10



for MNIST

Application to solvers for hard combinatorial problems (SAT, MIP, TSP)

- State-of-the-art solvers for NP-hard problems SAT, MIP, and TSP
- Between 4 and 76 parameters (choices of heuristics, etc)
- Performance highly dependent on these parameters
- Ran SMAC [Hutter et al, '11] ten times for each benchmark
  - achived speedups between 1.02x and 857x over default
  - fitted random forests on the union of the performance data
  - ran functional ANOVA on the random forest

Main effects explained a large fraction of variance: see table

Scenario	Raw Performance Main	Raw Performance Pairwise	Impr. over 25% quant. Main	Impr. over 25% quant. Pairwise	Impr. over def Main	Impr. over def Pairwise
SPEAR-BMC	88% (2s)	4% (112s)	50% (1s)	15% (36s)	26% (0s)	20% (23s)
SPEAR-SWV	76% (6s)	8% (348s)	19% (1s)	21% (80s)	74% (4s)	11% (250s)
CRYPTOMINISAT-BMC	28% (1s)	18% (62s)	31% (1s)	20% (39s)	6% (0s)	11% (5s)
CRYPTOMINISAT-SWV	37% (4s)	33% (182s)	9% (1s)	19% (44s)	24% (2s)	35% (70s)
SPARROW-3SAT1k	78% (0s)	15% (0s)	53% (0s)	31% (0s)	31% (0s)	34% (0s)
SPARROW-5SAT500	65% (0s)	28% (0s)	57% (0s)	34% (0s)	66% (0s)	27% (0s)
CAPTAINJACK-3SAT1k	42% (9s)	9% (1 321s)	21% (4s)	9% (599s)	37% (6s)	9% (832s)
CAPTAINJACK-5SAT500	20% (6s)	11% (917s)	18% (2s)	12% (308s)	26% (5s)	12% (726s)
SATENSTEIN-3SAT1k	45% (6s)	37% (845s)	23% (2s)	27% (296s)	27% (3s)	29% (334s)
SATENSTEIN-5SAT500	33% (9s)	45% (1 155s)	16% (3s)	32% (379s)	22% (5s)	49% (648s)
CPLEX-RCW	58% (5s)	6% (713s)	16% (1s)	33% (199s)	6% (1s)	15% (127s)
CPLEX-CORLAT	31% (29s)	7% (4 361s)	16% (10s)	22% (1 427s)	30% (22s)	16% (3 129s)
CPLEX-Regions200	61% (68s)	19% (10 416s)	26% (26s)	33% (3 476s)	13% (22s)	27% (2 787s)
CPLEX-CLS	55% (143s)	5% (21 502s)	2% (43s)	4% (5 725s)	5% (53s)	15% (6 047s)
CLASP-WeightedSeq	46% (13s)	13% (2 368s)	27% (5s)	20% (858s)	30% (6s)	20% (1047s)
CLASP-Riposte	39% (103s)	8% (18 518s)	10% (68s)	3% (12 213s)	15% (82s)	3% (1 4362s)

Main effect of Spear's most important parameter (its variable selection heuristic) on instances from hardware verification (left) and software verification (right)