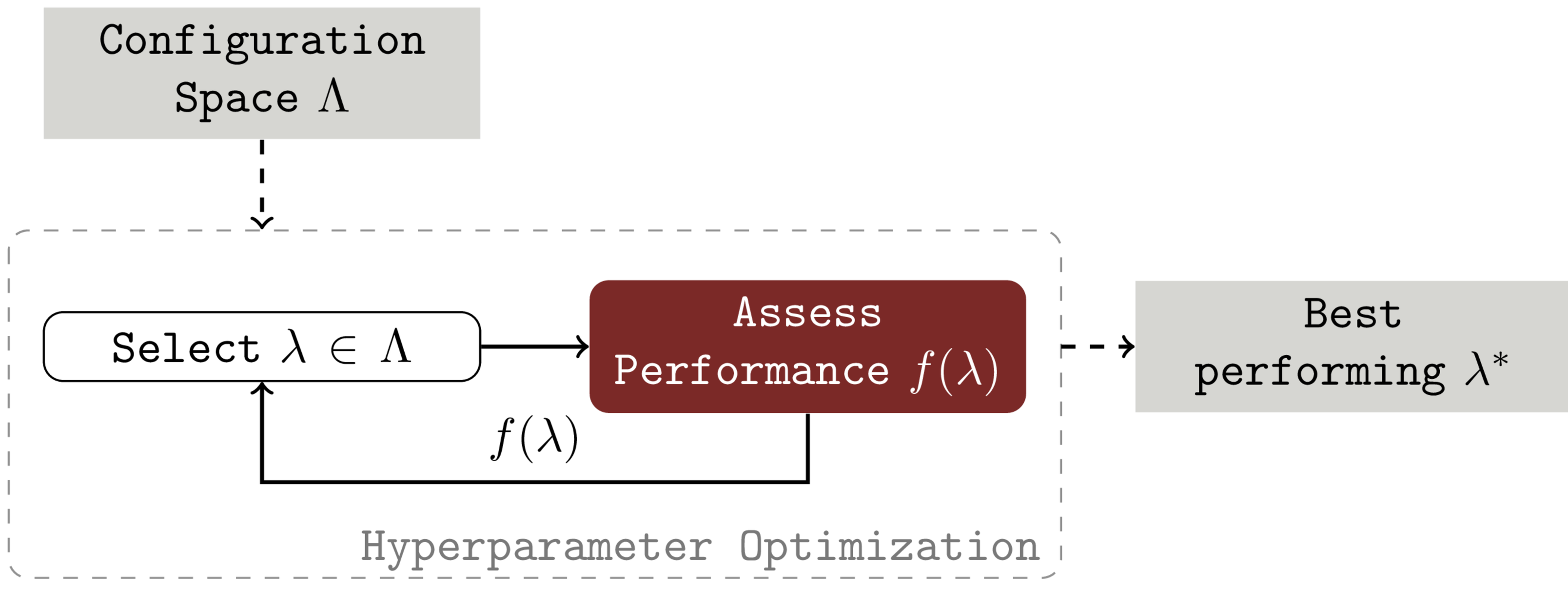
Efficient Benchmarking of Hyperparameter Optimizers via Surrogates

Katharina Eggensperger and Frank Hutter
University of Freiburg
{eggenspk,fh}@cs.uni-freiburg.de

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

…in 30 seconds

- **Benchmarking** hyperparameter optimization methods is costly, as it requires many evaluations of the used benchmark problem.
- We propose cheap, but realistic **surrogate benchmarks** based on predictive performance models.
- Our benchmarks are **publicly available** and allow extensive white-box tests and fast evaluation of optimization algorithms.



Our Approach

1. Collect <configuration, performance> pairs from different benchmarks
2. Train Regression model
3. Replace call to benchmark function with model prediction

Question: **How well does this work?**

Problems with existing benchmark functions

Realistic benchmark problems:

+ Complex & interesting
- Expensive to evaluate
- Complicated to set up (libraries, dependencies, special hardware, etc.)

Examples: Deep Neural Networks, onlineLDA, AutoWEKA

Synthetic test functions:

+ Easy to set up
+ Cheap to evaluate
- Unrealistic shape & too smooth

Examples: BBOB, Table-Lookups, Branin

Setup

8 regression models

- **Tree-based models**
- **Gaussian Processes**
- Support Vector Regression
- K-Nearest Neighbour
- Linear Regression

9 benchmark functions, e.g.

- Logistic Regression
- (Deep) Neural Networks
- onlineLDA

Experiments

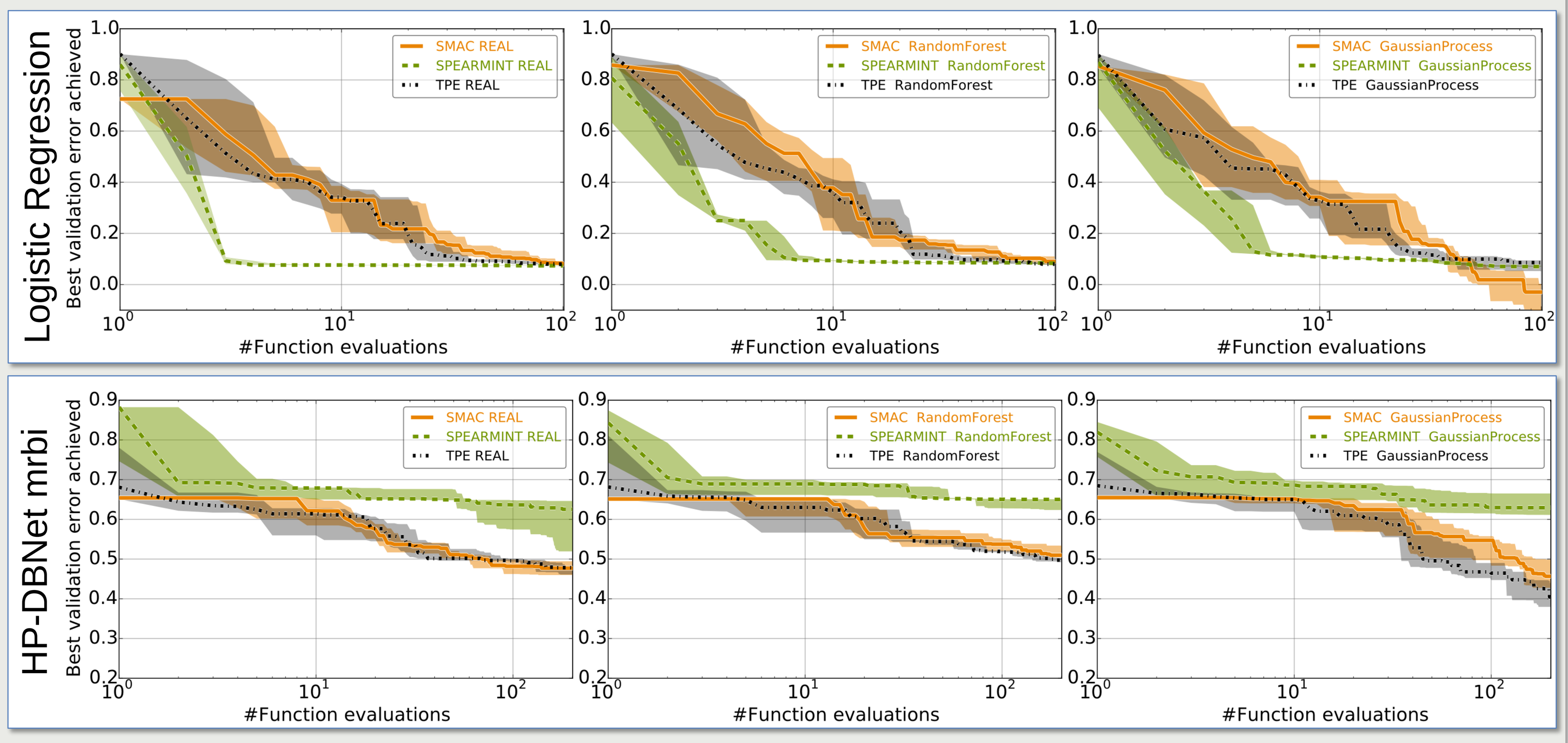
- Collect data by conducting **optimization runs** and **random search**
- Evaluate **quality** of regression models
- Compare optimizer performance on **true vs. surrogate benchmark**

Empirical Results

- Almost **perfect for low-dimensional** benchmarks and still **acceptable for higher dimensions**
- Reduce benchmark overhead to **<1 sec**

Answer: **Surrogates work well, especially based on Random Forests**

Figures: Best performance found by different optimizers over time. We show median and quartile across 10 runs on the real benchmark (left) and surrogate benchmarks based on random forests (middle) and Gaussian Processes (right).



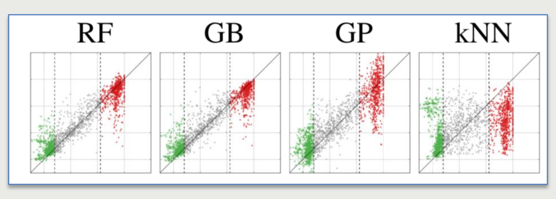


Figure: Performance of different regression models predicting configurations for optimizer SMAC

	#λ	hyperparameter cond.	hyperparameter cat. / cont.	One-hot dim.	#evals. per run	#data
onlineLDA	3	-	- / 3	4	50	1 999
Log. Reg.	4	-	- / 4	5	100	4 000
Log. Reg. 5CV				9	500	20 000
HP-NNET convex	14	4	7 / 7	25	200	8 000
HP-NNET convex 5CV				29	500	19 998
HP-NNET mrbi	14	4	7 / 7	25	200	8 000
HP-NNET mrbi 5CV				29	500	20 000
HP-DBNET convex	36	27	19 / 17	82	200	7 997
HP-DBNET mrbi						7 916

Table: Properties of the benchmarks for which we provide surrogate benchmarks

Experiment	#evals	Real: SMAC Valid. loss	Real: Spearmint Valid. loss	Real: TPE Valid. loss	RF-based surrogate: SMAC Valid. loss	RF-based surrogate: Spearmint Valid. loss	RF-based surrogate: TPE Valid. loss	GP-based surrogate: SMAC Valid. loss	GP-based surrogate: Spearmint Valid. loss	GP-based surrogate: TPE Valid. loss
Log.Reg.	100	0.08±0.00	**0.07**±0.00	0.08±0.00	0.10±0.02	**0.08**±0.00	0.08±0.01	**-0.06**±0.09	0.07±0.00	0.08±0.04
onlineLDA	50	1266.4±4.4	1264.3±4.9	**1263.7**±3.0	1268.2±2.0	**1265.6**±3.6	1266.1±2.0	1273.0±7.6	**1263.4**±4.5	1268.5±5.3
HP-NNET convex	200	0.19±0.01	0.20±0.01	**0.19**±0.01	**0.21**±0.01	0.21±0.00	0.21±0.01	0.14±0.03	0.14±0.05	**0.12**±0.05
HP-NNET mrbi	200	0.49±0.01	0.51±0.03	**0.48**±0.01	0.51±0.03	0.54±0.04	**0.49**±0.01	**0.47**±0.02	0.52±0.07	0.47±0.01
HP-DBNET convex	200	0.15±0.01	0.23±0.10	**0.15**±0.01	**0.18**±0.00	0.20±0.06	0.18±0.00	**0.11**±0.02	0.12±0.05	0.16±0.02
HP-DBNET mrbi	200	**0.47**±0.02	0.59±0.08	0.47±0.02	0.52±0.02	0.63±0.05	**0.50**±0.01	0.45±0.06	0.62±0.06	**0.41**±0.05
Log.Reg 5CV	500	**0.08**±0.00	0.08±0.00	0.09±0.01	0.08±0.00	**0.08**±0.00	0.09±0.01	**-0.04**±0.07	0.08±0.00	0.07±0.01
HP-NNET convex 5CV	500	**0.19**±0.01	0.23±0.05	0.21±0.01	**0.22**±0.01	0.24±0.04	0.22±0.01	0.17±0.01	**0.15**±0.05	0.18±0.02
HP-NNET mrbi 5CV	500	**0.48**±0.01	0.55±0.03	0.51±0.02	**0.51**±0.01	0.59±0.06	0.51±0.02	**0.38**±0.05	0.57±0.06	0.49±0.01

Table: Empirical comparison of three optimizers on various real and surrogate-based benchmarks

Benefit

Our surrogate benchmarks …

- provide a **60 to 3600 x speedup**
- require negligible computational resources
- allow **unit tests**
- help to **analyze** how optimizers work on complex benchmark

… and are **publicly** available:

For more information visit:
www.automl.org/benchmarks.html

SMAC: F. Hutter, H. Hoos, and K. Leyton-Brown, **Sequential model-based optimization for general algorithm configuration,** 2011
TPE: J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, **Algorithms for hyper-parameter optimization**, 2011
Spearmint: J. Snoek, H. Larochelle, and R. Adams, **Practical Bayesian optimization of machine learning algorithms**, 2012

HPOlib: K. Eggensperger, M. Feurer, F. Hutter, J. Bergstra, J. Snoek, H. Hoos, and K. Leyton-Brown, **Towards an empirical foundation for assessing Bayesian optimization of hyperparameters**, 2013

Implementation publicly available: https://github.com/KEggensperger/SurrogateBenchmarks