Speedup Techniques for Hyperparameter Optimization

Success Stories and Practical Recommendations

Bernd Bischl Frank Hutter Lars Kotthoff
Marius Lindauer Joaquin Vanschoren

Large-scale Meta-Learning for HPO in Industry (Facebook)

- Facebook has an internal self-service machine learning (ML) system
  - Non-ML departments can integrate highly optimized ML models into their workflow
  - Hyperparameters of the ML models are optimized with Bayesian optimization

Large-scale Meta-Learning for HPO in Industry (Facebook)

- Facebook has an internal self-service machine learning (ML) system
  - Non-ML departments can integrate highly optimized ML models into their workflow
  - Hyperparameters of the ML models are optimized with Bayesian optimization

- Training data for the models changes over time
  - Hyperparameters are constantly re-optimized
  - For efficiency: meta-learning Bayesian optimization, as described in [Feurer et al. 2018]

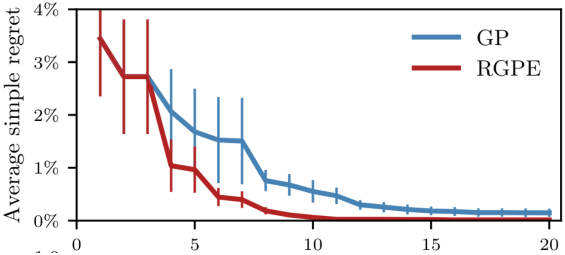
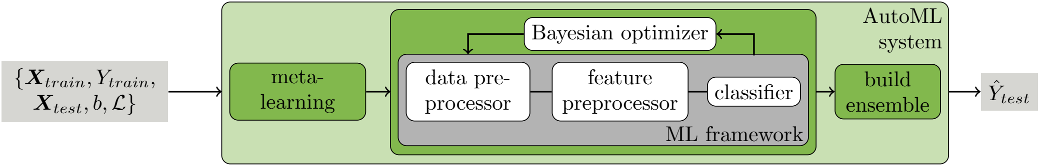


Figure: Bayesian optimization with meta-learning (RGPE) vs. vanilla Bayesian optimization (GP)

Auto-sklearn [Feurer et al. 2015]

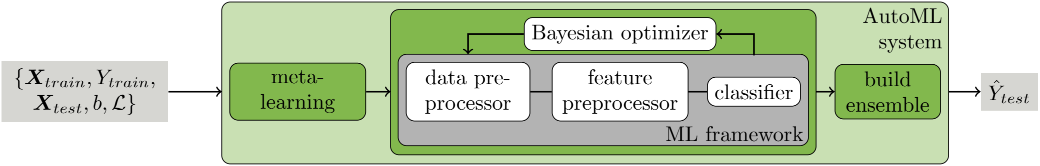
Extension of Auto-WEKA with focus on speed improvements and robustness:



- Uses meta-learning to warmstart Bayesian optimization
- Won the 1st AutoML challenge

Auto-sklearn [Feurer et al. 2015]

Extension of Auto-WEKA with focus on speed improvements and robustness:



- Uses meta-learning to warmstart Bayesian optimization
- Won the 1st AutoML challenge
- Open source (BSD) and trivial to use


```
>>> import autosklearn.classification
>>> cls = autosklearn.classification.AutoSklearnClassifier()
>>> cls.fit(X_train, y_train)
>>> predictions = cls.predict(X_test)
```

Available at `https://automl.github.io/auto-sklearn`; frequently used in industry

BOHB [Falkner et al. 2018]

- Robust and efficient
- Only published in 2018, adopted by the community very quickly

Cited by 129

Scholar articles

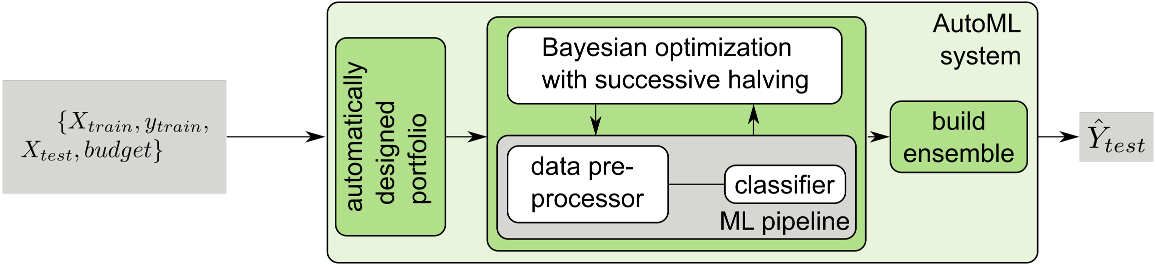
BOHB: Robust and efficient hyperparameter optimization at scale
S Falkner, A Klein, F Hutter - arXiv preprint arXiv:1807.01774, 2018
Cited by 129 Related articles All 8 versions

- Available at `https://github.com/automl/HpBandSter`

Unwatch 23 | Star 371 | Fork 80

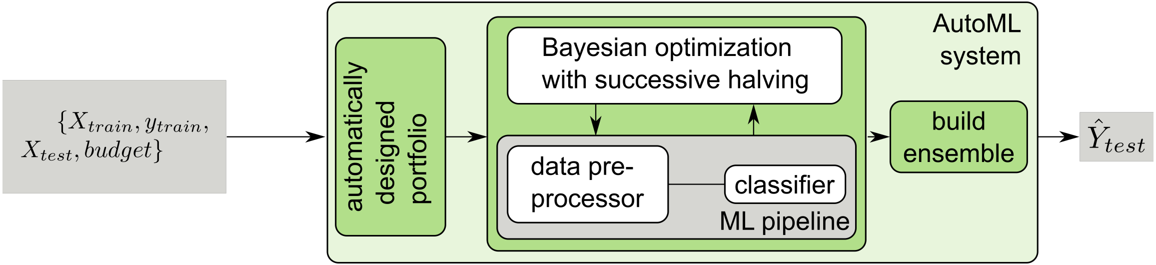
PoSH-Auto-sklearn [Feurer et al. 2018]

Idea: integrate warmstarting and a BOHB-like approach for Auto-sklearn



PoSH-Auto-sklearn [Feurer et al. 2018]

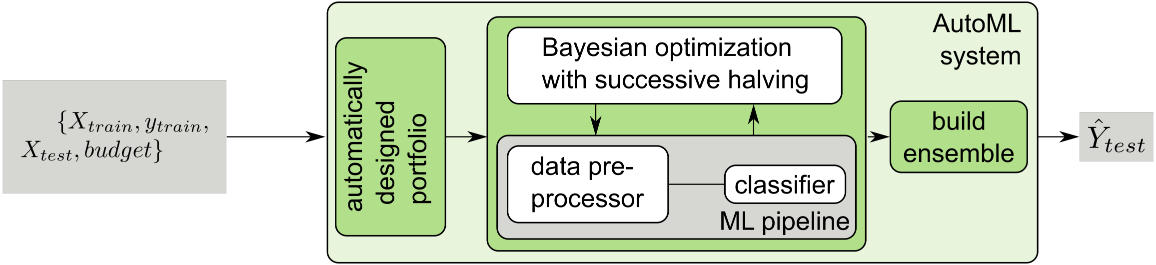
Idea: integrate warmstarting and a BOHB-like approach for Auto-sklearn



- Uses task-independent meta-learning to warmstart Bayesian optimization
  - Therefore, no need for (potentially unreliable) meta-features

PoSH-Auto-sklearn [Feurer et al. 2018]

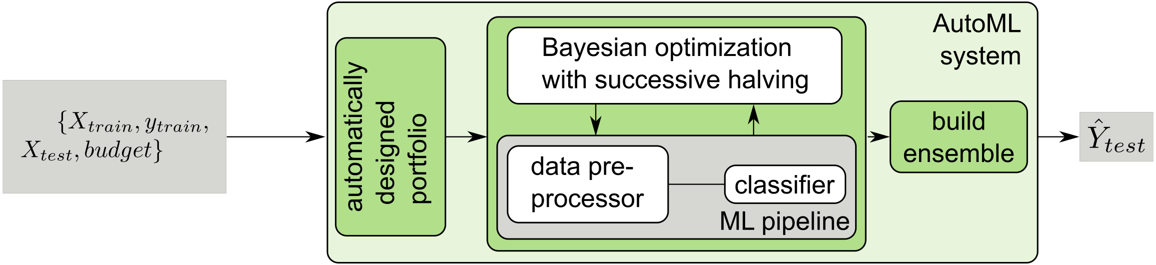
Idea: integrate warmstarting and a BOHB-like approach for Auto-sklearn



- Uses task-independent meta-learning to warmstart Bayesian optimization
  - Therefore, no need for (potentially unreliable) meta-features
- Uses successive halving to quickly go through proposed configurations
  - Therefore, scales better to larger datasets

PoSH-Auto-sklearn [Feurer et al. 2018]

Idea: integrate warmstarting and a BOHB-like approach for Auto-sklearn



- Uses task-independent meta-learning to warmstart Bayesian optimization
  - Therefore, no need for (potentially unreliable) meta-features
- Uses successive halving to quickly go through proposed configurations
  - Therefore, scales better to larger datasets
- Followed by BOHB-like approach (uses successive halving instead of Hyperband)
- Won the 2nd AutoML challenge

Auto-sklearn 2.0

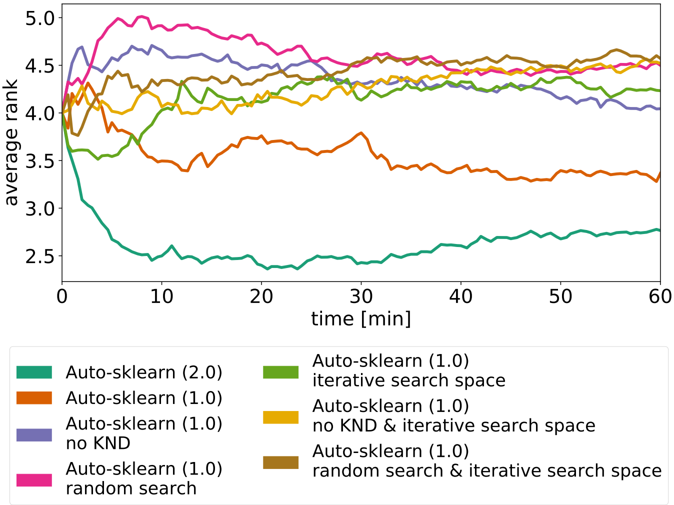
- Idea: automatically choose on a per-dataset basis
  - holdout or cross-validation
  - optimization on the full budget or optimization with successive halving

Auto-sklearn 2.0

- Idea: automatically choose on a per-dataset basis
  - holdout or cross-validation
  - optimization on the full budget or optimization with successive halving
- Can be done based on algorithm selection

Auto-sklearn 2.0

- Idea: automatically choose on a per-dataset basis
  - holdout or cross-validation
  - optimization on the full budget or optimization with successive halving
- Can be done based on algorithm selection
- Substantial improvements over Auto-sklearn 1.0
  - $5\times$ reduction of average error
  - $6\times$ speedup (same performance in 10 minutes as Auto-sklearn 1.0 in 1 hour)



Practical Recommendations Which HPO Method to Use [Feurer and Hutter. 2019]

- If multiple fidelities available: BOHB [Falkner et al. 2018]
- Otherwise
  - Low-dimensional continuous parameter space:
    - GP-based BO, e.g., Spearmint [Snoek et al. 2012]
  - High-dimensional discrete parameter space:
    - RF-based BO, e.g., SMAC [Hutter et al. 2011]
  - Purely continuous, cheap function evaluations:
    - CMA-ES [Hansen et al. 2001]; evaluated for HPO by [Loshchilov and Hutter. 2016]

Practical Recommendations Which HPO Method to Use [Feurer and Hutter. 2019]

- If multiple fidelities available: BOHB [Falkner et al. 2018]
- Otherwise
  - Low-dimensional continuous parameter space:
    - GP-based BO, e.g., Spearmint [Snoek et al. 2012]
  - High-dimensional discrete parameter space:
    - RF-based BO, e.g., SMAC [Hutter et al. 2011]
  - Purely continuous, cheap function evaluations:
    - CMA-ES [Hansen et al. 2001]; evaluated for HPO by [Loshchilov and Hutter. 2016]
- Just submitted: DEHB combines differential evolution and Hyperband and largely dominates BOHB. Especially good for high dimensions.

Questions to Answer for Yourself / Discuss with Friends

- Repetition. Discuss several success stories of speeding up Bayesian optimization.
- Repetition. What differs between Auto-sklearn 1.0 and Auto-sklearn 2.0?