Methods for Improving Bayesian Optimization for AutoML

Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Tobias Springenberg, Manuel Blum, Frank Hutter
{ feurerm | kleinaa | eggenspk | springj | blum | fh }@cs.uni-freiburg.de
Department of Computer Science, University of Freiburg, Germany

UNI FREIBURG

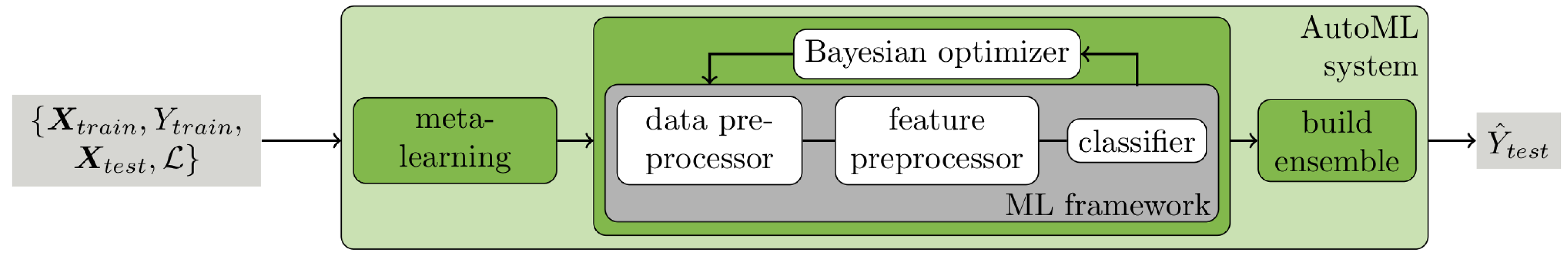
Basics

… in 30 seconds

- **AutoML**: choosing an algorithm and setting its hyperparameters for a new problem without human intervention
- Auto-WEKA showed the potential of combining WEKA and Bayesian optimization
- We do this for scikit-learn: **auto-sklearn**
- We extend this approach with two new components to **speed up convergence** (meta-learning) and improve **robustness** (ensemble learning)
- An early version of this work **won the auto track** of the first phase of the ongoing *ChaLearn AutoML challenge*

The AutoML Workflow

Previous state-of-the-art AutoML system (inner green box) together with our extensions (outer green box)



auto-sklearn

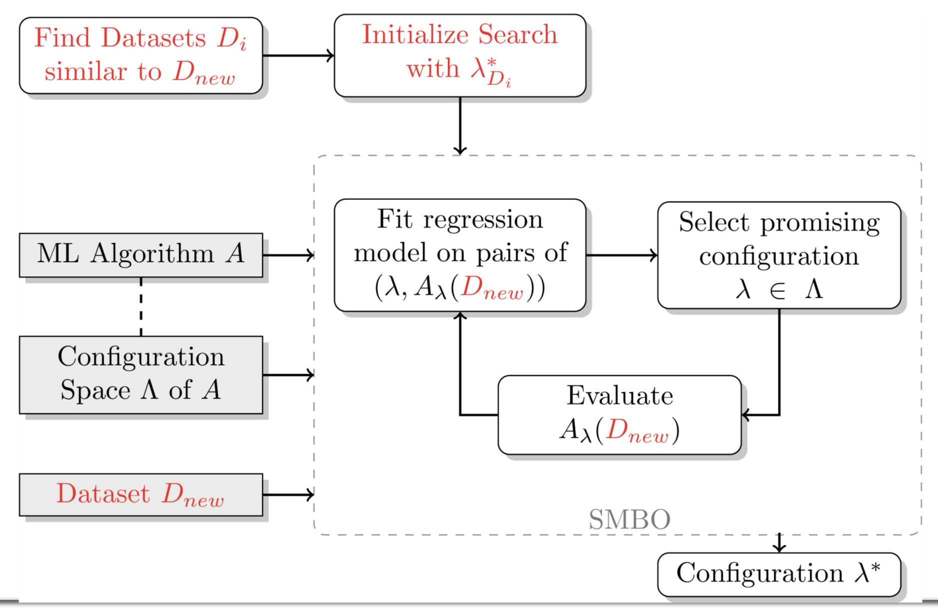
Machine Learning Pipeline

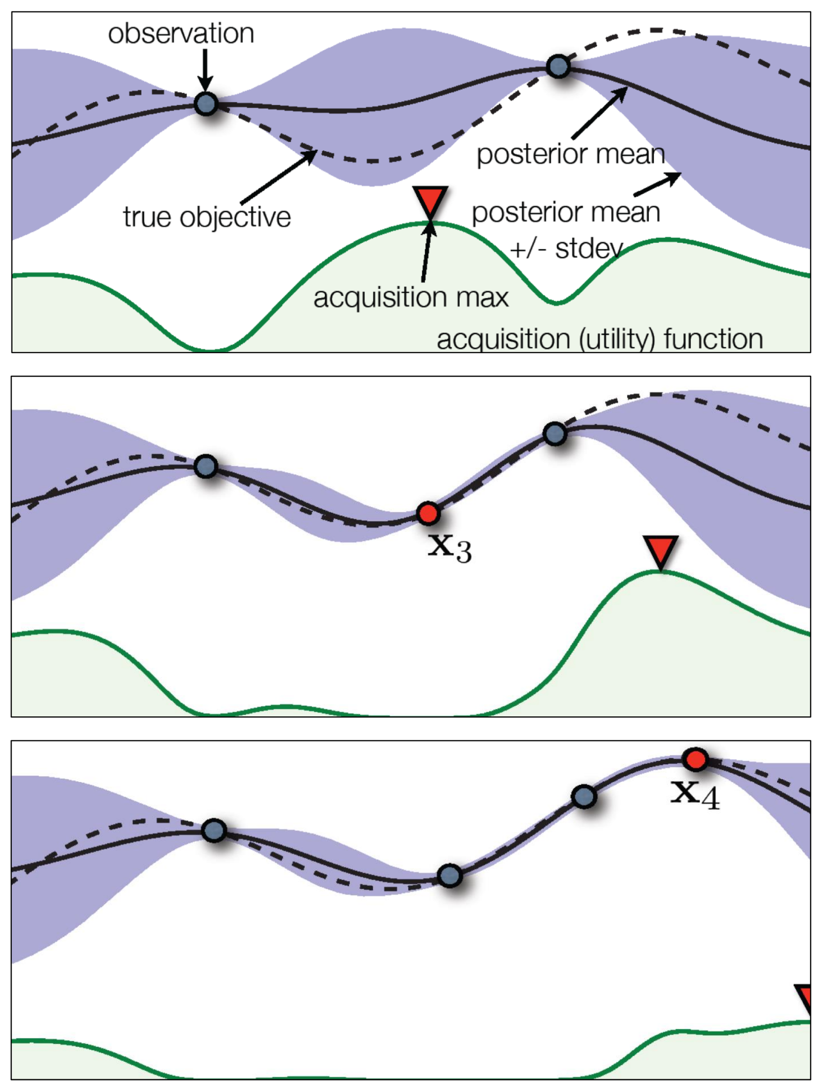
- A configurable machine learning pipeline built around scikit-learn
- We use 16 classifiers, 14 feature preprocessing methods and 3 data preprocessing methods; yielding a **C**ombined **A**lgorithm **S**election and **H**yperparameter Optimization (CASH) problem with 132 hyperparameters
- We use the Bayesian optimization toolkit SMAC to find good instantiations

Classifier	#λ	cat	(cond)	cont	(cond)	Feature Preprocessor	#λ	cat	(cond)	cont	(cond)
AdaBoost	3	-	-	3	(-)	Extremely rand. Trees	5	2	(-)	3	(-)
Bernoulli Naïve Bayes	2	1	(-)	1	(-)	Fast ICA	4	3	(-)	1	(-)
Decision Tree	3	1	(-)	2	(-)	Feature Agglomeration	3	2	(-)	1	(-)
Extremely rand. Trees	5	2	(-)	3	(-)	Kernel PCA	5	1	(-)	4	-3
Gaussian Naïve Bayes	-	-	(-)	-	(-)	Random Kitchen Sinks	2	-	(-)	2	(-)
Gradient Boosting	6	-	(-)	6	(-)	Linear SVM	5	3	(-)	2	(-)
kNN	3	2	(-)	1	(-)	No Preprocessing	-	-	(-)	4	(-)
LDA	2	-	(-)	2	(-)	Nystroem Sampler	5	1	(-)	1	-3
Linear SVM	5	3	(-)	2	(-)	PCA	2	1	(-)	4	(-)
Kernel SVM	8	3	(-)	5	2	Random Trees Embedding	4	-	(-)	1	(-)
Multinomial Naïve Bayes	2	1	(-)	1	(-)	Select Percentile	2	1	(-)	1	(-)
Passive Aggressive	3	1	(-)	2	(-)	Select Rates	3	2	(-)	1	(-)
QDA	2	-	(-)	2	(-)	Data preprocessor					
Random Forest	5	2	(-)	3	(-)	Imputation	1	-	(-)	1	(-)
Ridge Regression	2	-	(-)	2	(-)	Balancing	1	-	(-)	1	(-)
SGD	9	3	(-)	6	3	Rescaling	1	-	(-)	1	(-)

Meta-learning & Bayesian Optimization

- Bayesian optimization has to explore a very large configuration space
- We use meta-learning to initialize Bayesian optimization
- Distance between datasets is the L_1-distance of their meta-features
- For a new dataset, we start Bayesian optimization with configurations that worked best on the most similar datasets





Ensemble Learning

- Ensembles almost always outperform single models
- Bayesian Optimization throws away many trained models (wasteful)
- After each evaluation of a machine learning model we save its validation prediction
- We used the ensemble selection (ES) method by Rich Caruana et al. to build an ensemble based on the model's validation prediction after SMAC finished
- To optimize the single models' weights, ES starts from an empty set E and greedily adds models to E (with uniform weight, but allowing for repetitions) to optimize ensemble performance

Experiments

Vanilla auto-sklearn vs. Auto-WEKA

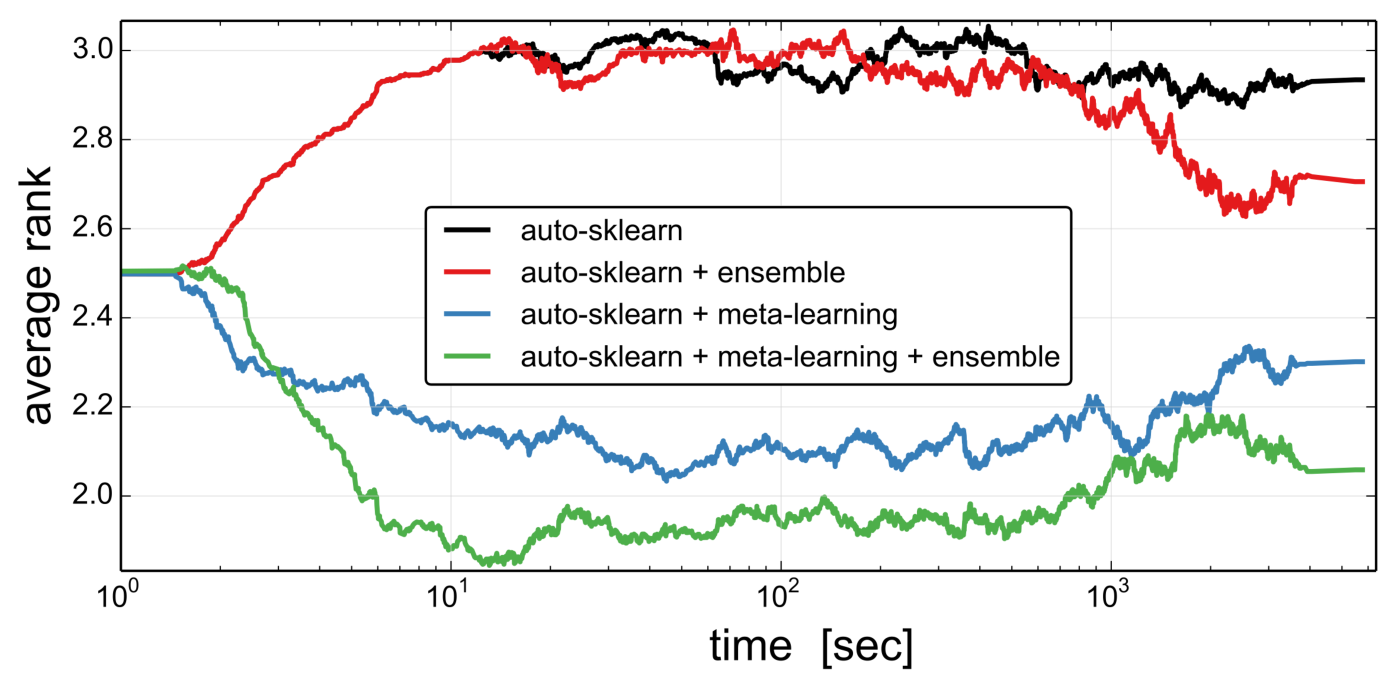
- Comparison using the original Auto-WEKA setup:
  - Test performance of the best configuration found with 10-fold cross-validation
  - Used 30 hours and 3GB RAM to search for the best configuration
- Vanilla auto-sklearn performs significantly better in 12/21 cases, ties in 5/21 and looses in 4/21

	Abalone	Amazon	Car	Cifar10	Cifar10 Small	Convex	Dexter	Dorothea	German Credit	Gisette	KDD09 Appetency
Auto-WEKA	**73.50**	30.00	**0.00**	61.47	56.19	21.49	**5.56**	**5.22**	28.00	2.24	**1.74**
Vanilla auto-sklearn	80.20	**13.99**	0.19	**51.93**	**52.28**	**14.95**	7.78	5.51	**26.00**	**1.29**	**1.74**

	KR-vs-KP	Madelon	MNIST Basic	MRBI	Secom	Semeion	Shuttle	Waveform	Wine Quality	Yeast
Auto-WEKA	**0.31**	19.62	**2.84**	59.85	**7.87**	**4.82**	**0.01**	14.20	**33.22**	**37.08**
Vanilla auto-sklearn	0.42	**12.82**	2.87	**47.84**	**7.87**	5.03	**0.01**	**14.07**	35.16	38.65

Evaluation of our extensions to AutoML

- Setup:
  - Ran auto-sklearn for 1 hour to simulate the AutoML challenge setting
  - Tested four different versions of auto-sklearn
  - Used 140 datasets from `OpenML.org`, each with at least 1000 samples
  - Leave-one-dataset-out: ran auto-sklearn on one dataset and assumed knowledge of all other 139.
- Both meta-learning and ensemble building improve auto-sklearn; auto-sklearn is further improved when both methods are combined.



Alpha version publicly available: https://github.com/automl/auto-sklearn