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

... 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

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 Combinatorial Algorithm Selection and Hyperparameter Optimization (CASH) problem with 132 hyperparameters
- We use the Bayesian optimization toolkit SMAC to find good instantiations

<table>
<thead>
<tr>
<th>Classifier</th>
<th>µs</th>
<th>cat</th>
<th>cont</th>
<th>cont</th>
<th>Feature Preprocessor</th>
<th>µs</th>
<th>cat</th>
<th>cont</th>
<th>cont</th>
<th>Feature Preprocessor</th>
</tr>
</thead>
<tbody>
<tr>
<td>KNN</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>No Preprocessing</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>No Preprocessing</td>
</tr>
<tr>
<td>QDA</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Hyperplane Sample</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Hyperplane Sample</td>
</tr>
<tr>
<td>Linear SVM</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>No Preprocessing</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>No Preprocessing</td>
</tr>
<tr>
<td>LogisticRegression</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>No Preprocessing</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>No Preprocessing</td>
</tr>
<tr>
<td>RandomForest</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>No Preprocessing</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>No Preprocessing</td>
</tr>
<tr>
<td>RidgeRegression</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>No Preprocessing</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>No Preprocessing</td>
</tr>
<tr>
<td>SGD</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>No Preprocessing</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>No Preprocessing</td>
</tr>
</tbody>
</table>

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

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

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

<table>
<thead>
<tr>
<th>Method</th>
<th>R2</th>
<th>MSE</th>
<th>RMSE</th>
<th>MAE</th>
<th>R2adj</th>
<th>MSEadj</th>
<th>RMSEadj</th>
<th>MAEadj</th>
<th>Weight</th>
<th>Quality</th>
<th>Yeast</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto-WEKA</td>
<td>73.50</td>
<td>30.00</td>
<td>0.00</td>
<td>61.47</td>
<td>56.19</td>
<td>21.49</td>
<td>5.56</td>
<td>5.22</td>
<td>28.00</td>
<td>2.24</td>
<td>1.74</td>
</tr>
<tr>
<td>vanilla auto-sklearn</td>
<td>80.20</td>
<td>13.99</td>
<td>0.19</td>
<td>51.93</td>
<td>52.28</td>
<td>14.95</td>
<td>7.78</td>
<td>5.51</td>
<td>26.00</td>
<td>1.29</td>
<td>1.74</td>
</tr>
</tbody>
</table>

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