PADERBORN UNIVERSITY

PADERBORN UNIVERSITY

# OPTIMIZER ENSEMBLES FOR AUTOMATED MACHINE LEARNING

## MASTER THESIS DEFENSE

Lukas Brandt

Paderborn, 07.10.2020

# Content

Optimizer Ensembles for Automated Machine Learning

# Content

$\underbrace{\text{Optimizer Ensembles}}$ for $\underbrace{\text{Automated Machine Learning}}$

1 AutoML:

- What?
- Why?
- How?
- Problems?

# Content

$\underbrace{\text{Optimizer Ensembles}}_{\text{Approach of this thesis}}$ for $\underbrace{\text{Automated Machine Learning}}_{}$

1 AutoML:

- What?
- Why?
- How?
- Problems?

# Content

$\underbrace{\text{Optimizer Ensembles}}_{\text{Approach of this thesis}}$ for $\underbrace{\text{Automated Machine Learning}}_{\text{AutoML}}$

Approach of this thesis

2 How?

1 AutoML:

- What?
- Why?
- How?
- Problems?

# Content

Optimizer Ensembles for Automated Machine Learning

Approach of this thesis

2. How?
3. Does it work?

1. AutoML:
   - What?
   - Why?
   - How?
   - Problems?

# Content

$\underbrace{\text{Optimizer Ensembles}}_{}$ for $\underbrace{\text{Automated Machine Learning}}_{}$

Approach of this thesis

2 How?

3 Does it work?

4 In conclusion?

1 AutoML:

- What?
- Why?
- How?
- Problems?

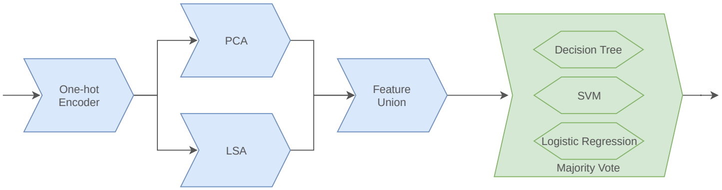
# AutoML – What?

Tool to assist in machine learning / data science use-cases:

- User provides data in machine-readable format
- AutoML tool tries to find the best suited machine learning pipeline and parametrization for this data within a budget constraint

# AutoML – Why?

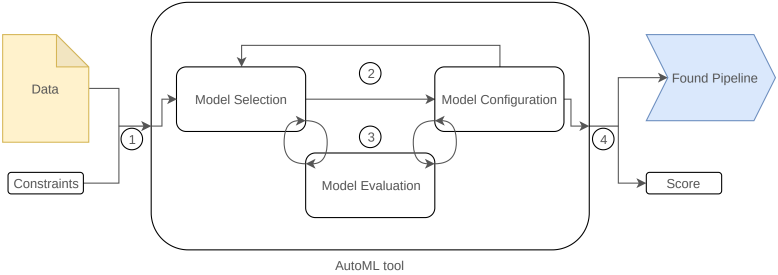
For complex datasets, more sophisticated pipelines might be necessary:




→ Domain knowledge and manual work would required for construction and configuration of pipelines without AutoML

# AutoML – How?

Workflow of a simplified AutoML tool:



# AutoML – How?

Properties of the AutoML problem:

- Model selection and model configuration usually have enormous candidate spaces
- Relationship between candidates and evaluation scores is not analytically solvable
- Ideally, the next candidate selection should be based on the knowledge aggregated during previous model selection + model configuration iterations

→ Black-box optimization problem

# AutoML – How?

Approach of several established state-of-the-art AutoML tools:

- Combine model selection and model configuration spaces into one optimization space
- To make this feasible, allowed pipeline topologies are often heavily restricted
- A candidate of the space encodes the selected pipeline components as well as the configuration
- Utilize one black-box optimization algorithm for this combined space, e.g. Genetic Algorithms or Bayesian Optimization

# AutoML – Problems?

1. **Optimization space dimensionality**:
   - Dimensionality of model configuration space depends on the result of the model selection
   - Optimization in a space with changing dimensionality is really hard/impossible

Thus, dimensionality of combined space has to be pre-definied:

- Use a low dimensionality → Maximum length of candidate pipelines and amount of parameters available for configuration is restricted
- Use a high dimensionality → Optimization algorithms will struggle

# AutoML – Problems?

**2. No-Free-Lunch theorems:**

*"[...] if an algorithm does particularly well on average for one class of problems then it must do worse on average over the remaining problems"* [1]

→ A single black-box optimization algorithm cannot be optimal for all AutoML problem classes (dataset, timeout, ...)

[1] D. H. Wolpert and W. G. Macready. "No free lunch theorems for optimization" (1997)

# AutoML – Problems?

Two recently published approaches started to tackle these problems:

- *ReinBo*: Model selection via reinforcement learning and decoupled model configuration via Bayesian optimization
- *Mosaic*: Model selection via MCTS and decoupled model configuration via Bayesian optimization

**Dimensionality**: Pipelines still have a maximum length and can only be linear, but the decoupled model configuration spaces can have an arbitrary dimensionality

**No-Free-Lunch theorems**: Two different optimization algorithms were applied in one approach. Thus, the tool is less restricted by the suitability of one optimization algorithm

# Optimizer Ensembles – How?

**Contribution of this thesis:**

- Development of a method for AutoML via optimizer ensembles to approach both problems, caused by dimensionality and the No-Free-Lunch theorems, at once
- Model selection in this method should not restrict the pipeline length or allowed topology complexity
- Empirical evaluation of the approach

# Optimizer Ensembles – How?

**Dimensionality**

Partition into model selection and model configuration spaces:

1. Reduce combined space to a model selection problem
2. Create model configuration spaces individually deduced for each model selection result on-the-fly

$\rightarrow$ Pipeline topologies of the model selection space can have arbitrary length and complexity

# Optimizer Ensembles – How?

**No-Free-Lunch theorems**:
Integrate several optimization algorithms into an ensemble to explore all and exploit the best suited one

→ Since multiple optimizers work in the same space, use the priorly gathered knowledge about the optimization space landscape (warmstarting)

# Optimizer Ensembles – How?

Let a pipeline topology *p* be given by some model selection method:

- Best suited optimizer has to be chosen for a model configuration of *p*
- Suitability could be influenced by several factors:
  - Input dataset
  - Model configuration optimization landscape for the components of *p*
  - Presented warmstarting data
- Suitability is unknown without a large study beforehand

# Optimizer Ensembles – How?

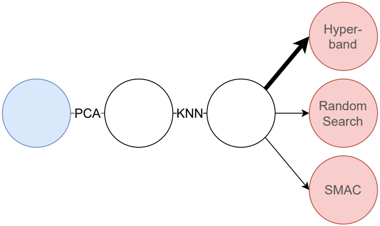
**Optimizer selection:**

- Intelligent try-and-error approach is required
- Formalize turn-wise optimizer choice as a Multi-Armed Bandit problem
- Within a sufficient number of turns, most Multi-Armed Bandit algorithms should be able to balance exploration and exploitation of optimizers to exploit the best suited one sufficiently

# Optimizer Ensembles – How?

For example:

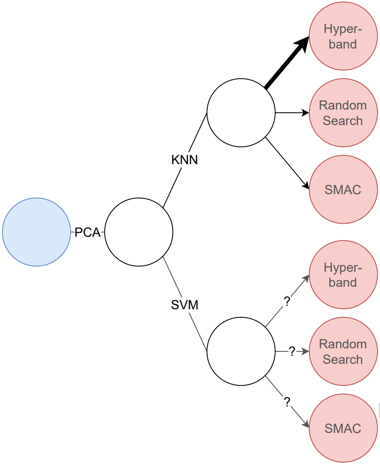
Hyperband might be exploited the most for a PCA + KNN pipeline



# Optimizer Ensembles – How?

If PCA + KNN + Hyperband yielded good results, could for example PCA be a good pre-processor choice in general?

→ If this can be assumed, other combinations of PCA with classifier and model configuration optimizer should be explored as well



# Optimizer Ensembles – How?

- A Multi-Armed Bandit algorithm that is designed for these assumed hierarchical dependencies is the UCT algorithm, as for instance applied in an MCTS
- Now, MCTS could be used to combine model selection and selecting an optimizer for the model configuration of each model selection result into a single task

# Optimizer Ensembles – How?

The search graph has the following structure:

- Create a model selection search graph as an HTN planning graph (see ML-Plan approach[1])
- This graph has leaf nodes, where a complete pipeline topology can be deduced (*Pipeline nodes*)
- One child node for each optimization algorithm is attached to all pipeline nodes (*Optimizer nodes*)

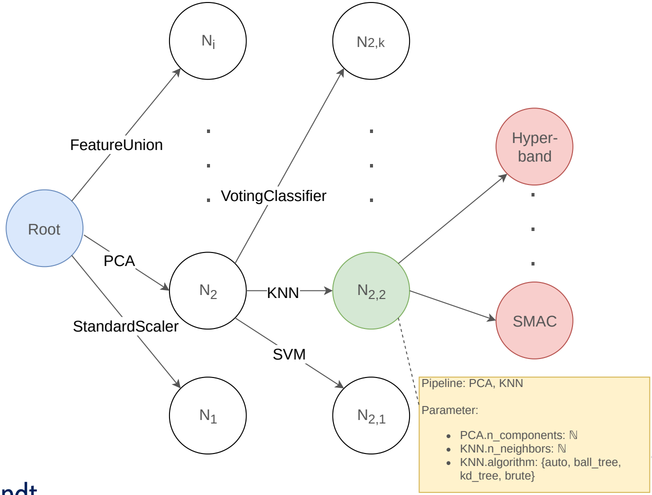
[1] F. Mohr, M. Wever, E. Hüllermeier. "ML-Plan: Automated machine learning via hierarchical planning" (2018)

# Optimizer Ensembles – How?

This graph is now searched with an MCTS in the following manner:

- Pipeline nodes store model configuration space of their pipeline and results of previous evaluations for warmstarting
- MCTS expands optimizer node or Monte-Carlo simulation ends in optimizer node → Model configuration Optimization run with a certain optimization budget is started
- AutoML time budget is spent → Best found pipeline topology + parametrization is returned as a result

# Optimizer Ensembles – How?



# Optimizer Ensembles – How?

In the context of this thesis, *Optimizer Ensembles* are a combination of:

- Different optimization algorithms explored and exploited as a hierarchical Multi-Armed Bandit problem for all pipelines
- Sharing of gathered optimization landscape knowledge and utilizing it via warmstarting

# Optimizer Ensembles – Does it work?

- A reference implementation was developed in Python for an evaluation
- Pipelines are constructed from scikit-learn components
- Utilized optimizers were: Genetic Algorithm, SMAC, Hyperband, Random Search, Discretization Search (i.e. model configuration method of ML-Plan)

# Optimizer Ensembles – Does it work?

Reference implementation is used to answer two research questions:

1. Is this approach competitive with other state-of-the-art approaches?

2. Based on the idea of automatically exploring and exploiting the suitability of optimizers, can knowledge be extrapolated from this?

# Optimizer Ensembles – Does it work?

1. State-of-the-art benchmark

# Optimizer Ensembles – Does it work?

- 9 Datasets (Car, Cifar-10, Dexter, Dorothea, Kr-vs-Kp, Semeion, Waveform, Wine quality, Yeast)
- 30 Random seeds
- AutoML timeout: 1h
- Node evaluation timeout: 5 Minutes
- Benchmark approaches:
    - `auto-sklearn`
    - `ML-Plan`
    - `Mosaic`
    - `TPOT`
- AutoML solution space: auto-sklearn space

# Optimizer Ensembles – Does it work?

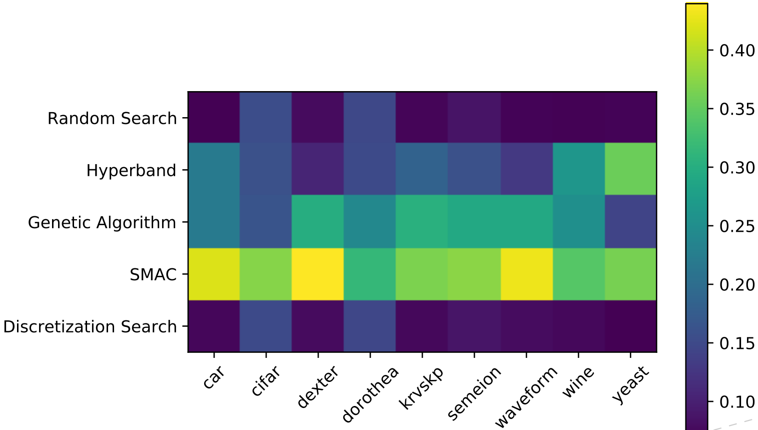
Analysis of the benchmark approaches in regards of significant improvements or deteriorations via Welch's $t$-tests with $p = 0.05$

| | *Improvement* | *Deterioration* | *No significant difference* |
|---|---|---|---|
| `auto-sklearn` | 7 | 0 | 2 |
| `ML-Plan` | 4 | 5 | 0 |
| `Mosaic` | 6 | 0 | 3 |
| `TPOT` | 4 | 1 | 4 |

# Optimizer Ensembles – Does it work?

2. Optimizer suitability

# Optimizer Ensembles – Does it work?



# Optimizer Ensembles – Does it work?

Correlations between dataset properties and optimizer utilization for all datasets

Dataset properties:

- Instances
- Features
- Classes
- Dimensionality
- Autocorrelation
- Minority Class
- Majority Class

Optimizer utilization metrics:

- Relative
- Absolute
- Ranking

Coefficient of Spearman's rank correlation: $-0.1662 \leq \rho \leq 0.2286$

# Optimizer Ensembles – In conclusion?

MCTS model selection in an HTN planning graph + Optimizer Ensemble model configuration with warmstarting was partially competitive

↓

It could close the gap to the current state-of-the-art with improvements in the form of future work and additional research

# Optimizer Ensembles – In conclusion?

Possible future work for improvements:

- Adjust configuration dynamically during the search (Monte-Carlo simulations, Optimizer time budget, warmstarting, ...)
- Evaluate different MCTS policies or other search algorithms
- Evaluate different optimization algorithms to form the ensemble

## Thank you for your attention!

## Any questions or feedback?