Evolutionary optimization of machine learning pipelines

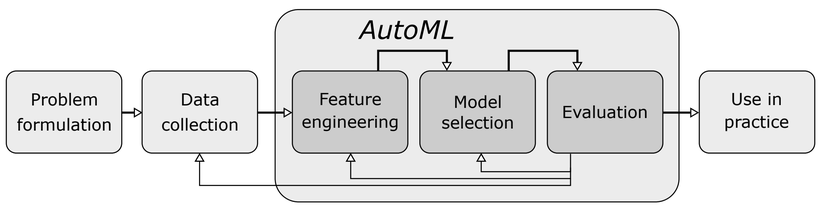
Gabriela Suchopárová

Abstract

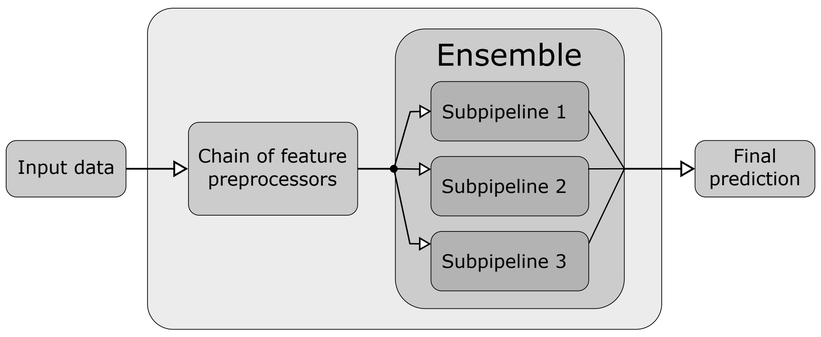
The subject of this work is the automated machine learning (AutoML), which is a field that aims to automatize the process of model selection for a given machine learning problem. We have developed a system that, for a given supervised learning task represented by a dataset, finds a suitable pipeline — combination of machine learning, ensembles and preprocessing methods. For the search we designed a special instance of the developmental genetic programming which enables us to encode directed acyclic graph pipelines into a tree representation. The system is implemented in the Python programming language and operates on top of the scikit-learn library. The performance of our solution was tested on 72 datasets of the OpenML-CC18 benchmark with very good results.

Workflows

Here will be a general workflow description.



Existing systems focused only on... The subject of this type of AutoML are mainly pipelines. Represented by a DAG.



Developmental GP

For pipeline optimization, we used the genetic programming (GP), which is a subfield of evolutionary algorithms. An individual in GP is in fact a computer program. Usually, it is represented as an expression tree. The fitness of the tree is determined by running the program, or also evaluating all functions and their arguments.

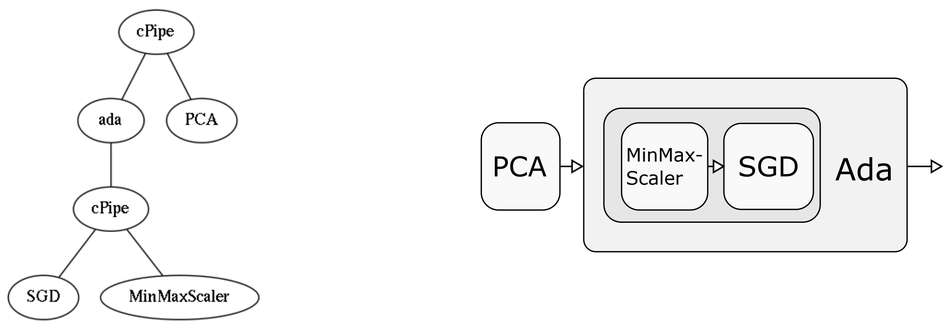


Figure 1: Example of an encoded pipeline

We created a specific encoding that enables to convert pipelines in the form of a DAG into a tree representation. Instead of directly encoding pipeline steps as nodes, we apply the developmental GP, where the nodes represent *operations* that create the pipeline.
An example of the encoding is shown in Figure 1. The tree individual contains instructions which construct the actual pipeline:

cPipe — create a pipeline with a preprocessor chain and an estimator
- **ada** — insert an AdaBoost ensemble with a base-estimator
  - **cPipe** — create a pipeline with a preprocessor chain
    - **SGD** — insert a SGD classifier
    - **MinMaxScaler** — insert a MinMaxScaler
- **PCA** — insert the PCA preprocessor

OpenML-CC18

Here will be a nice description of the experiment. May be reduced to only one of the graphs for illustrative purpose only.