AutoML: Dynamic Configuration & Learning

Overview

Bernd Bischl Frank Hutter Lars Kotthoff
Marius Lindauer Joaquin Vanschoren

Black vs. Grey vs. White Box

- Often we treat AutoML as a black-box problem
  - Black box: We choose input to the black box and observe outcome

Black vs. Grey vs. White Box

- Often we treat AutoML as a black-box problem
  - Black box: We choose input to the black box and observe outcome
  - E.g., classical hyperparameter optimizer:
    Input: Hyperparameter configuration $\rightarrow$ Output: Accuracy

Black vs. Grey vs. White Box

- Often we treat AutoML as a black-box problem
  - Black box: We choose input to the black box and observe outcome
  - E.g., classical hyperparameter optimizer:
    Input: Hyperparameter configuration $\rightarrow$ Output: Accuracy
- We discussed how to extend AutoML to a more grey-box approach:
  - Grey Box: We still choose the input, but we can observe more than the outcome, e.g, intermediate results

Black vs. Grey vs. White Box

- Often we treat AutoML as a black-box problem
  - Black box: We choose input to the black box and observe outcome
  - E.g., classical hyperparameter optimizer:
    Input: Hyperparameter configuration $\rightarrow$ Output: Accuracy
- We discussed how to extend AutoML to a more grey-box approach:
  - Grey Box: We still choose the input, but we can observe more than the outcome, e.g, intermediate results
  - We might can control the "box" a bit, e.g., early termination

Black vs. Grey vs. White Box

- Often we treat AutoML as a black-box problem
  - Black box: We choose input to the black box and observe outcome
  - E.g., classical hyperparameter optimizer:
    Input: Hyperparameter configuration $\rightarrow$ Output: Accuracy
- We discussed how to extend AutoML to a more grey-box approach:
  - Grey Box: We still choose the input, but we can observe more than the outcome, e.g, intermediate results
  - We might can control the "box" a bit, e.g., early termination
  - E.g., learning curve predictions, multi-fidelity optimization, ...

Black vs. Grey vs. White Box

- Often we treat AutoML as a black-box problem
  - Black box: We choose input to the black box and observe outcome
  - E.g., classical hyperparameter optimizer:
    Input: Hyperparameter configuration $\rightarrow$ Output: Accuracy
- We discussed how to extend AutoML to a more grey-box approach:
  - Grey Box: We still choose the input, but we can observe more than the outcome, e.g, intermediate results
  - We might can control the "box" a bit, e.g., early termination
  - E.g., learning curve predictions, multi-fidelity optimization, ...
  - $\rightsquigarrow$ often more efficient than black-box approaches (if done right)

Black vs. Grey vs. White Box

- Often we treat AutoML as a black-box problem
  - Black box: We choose input to the black box and observe outcome
  - E.g., classical hyperparameter optimizer:
    Input: Hyperparameter configuration $\rightarrow$ Output: Accuracy
- We discussed how to extend AutoML to a more grey-box approach:
  - Grey Box: We still choose the input, but we can observe more than the outcome, e.g, intermediate results
  - We might can control the "box" a bit, e.g., early termination
  - E.g., learning curve predictions, multi-fidelity optimization, ...
  - $\rightsquigarrow$ often more efficient than black-box approaches (if done right)
- Ultimately, we would like to treat AutoML as a white-box problem
  - White-box: We can observe and control all details of an algorithm run

Black vs. Grey vs. White Box

- Often we treat AutoML as a black-box problem
  - Black box: We choose input to the black box and observe outcome
  - E.g., classical hyperparameter optimizer:
    Input: Hyperparameter configuration → Output: Accuracy
- We discussed how to extend AutoML to a more grey-box approach:
  - Grey Box: We still choose the input, but we can observe more than the outcome, e.g, intermediate results
  - We might can control the "box" a bit, e.g., early termination
  - E.g., learning curve predictions, multi-fidelity optimization, . . .

  ⇝ often more efficient than black-box approaches (if done right)
- Ultimately, we would like to treat AutoML as a white-box problem
  - White-box: We can observe and control all details of an algorithm run

⇝ Goal: Replace algorithm components by learned policies

Iterative Optimization Heuristics

IOHs

Iterative Optimization Heuristics (IOHs) propose a set of solution candidates in each iteration based on previous evaluations.

Iterative Optimization Heuristics

IOHs

Iterative Optimization Heuristics (IOHs) propose a set of solution candidates in each iteration based on previous evaluations.

Important Observations:

- Many ML algorithms are iterative in nature, in particular for big data, e.g.:
  - SGD (for linear models or for deep neural networks)
  - Tree-based algorithms

Iterative Optimization Heuristics

IOHs

Iterative Optimization Heuristics (IOHs) propose a set of solution candidates in each iteration based on previous evaluations.

Important Observations:

- Many ML algorithms are iterative in nature, in particular for big data, e.g.:
  - SGD (for linear models or for deep neural networks)
  - Tree-based algorithms
- Often we have only a single solution candidate (e.g., weights of neural network)
  - If we use a evoluationary strategy as in neural evoluation, we have a population of solution candidates

Iterative Optimization Heuristics

IOHs

Iterative Optimization Heuristics (IOHs) propose a set of solution candidates in each iteration based on previous evaluations.

Important Observations:

- Many ML algorithms are iterative in nature, in particular for big data, e.g.:
  - SGD (for linear models or for deep neural networks)
  - Tree-based algorithms
- Often we have only a single solution candidate
  (e.g., weights of neural network)
  - If we use a evoluationary strategy as in neural evoluation,
    we have a population of solution candidates
- Hopefully, the quality of solution candiates improves in each iteration
  - Update of the weights of a neural network

Iterative Optimization Heuristics

IOHs

Iterative Optimization Heuristics (IOHs) propose a set of solution candidates in each iteration based on previous evaluations.

Important Observations:

- Many ML algorithms are iterative in nature, in particular for big data, e.g.:
  - SGD (for linear models or for deep neural networks)
  - Tree-based algorithms
- Often we have only a single solution candidate
  (e.g., weights of neural network)
  - If we use a evoluationary strategy as in neural evoluation,
    we have a population of solution candidates
- Hopefully, the quality of solution candiates improves in each iteration
  - Update of the weights of a neural network
- Main component is the heuristic for proposal mechanism of new solution candidates

Learning for IOHs

Dynamic Adaptation of Hyperparameters

The goal is to dynamically adapt hyperparameters based on some feedback from the algorithm.

Learning for IOHs

Dynamic Adaptation of Hyperparameters

The goal is to dynamically adapt hyperparameters based on some feedback from the algorithm.

Dynamic Algorithm Configuration: DAC

The goal of DAC is to learn a policy from data
that adapts the hyperparameter settings of an IOH.

Learning for IOHs

Dynamic Adaptation of Hyperparameters

The goal is to dynamically adapt hyperparameters based on some feedback from the algorithm.

Dynamic Algorithm Configuration: DAC

The goal of DAC is to learn a policy from data
that adapts the hyperparameter settings of an IOH.

Learning to Learn: L2L

The goal of L2L is to learn a proposal mechanism from data.