Solving ML with ML: Evaluating the performance of the Monte Carlo Tree Search algorithm in the context of Program Synthesis

Motivation

- The global machine learning market is expected to grow at 38.8% a year for the upcoming 6 years
- Pre- and post-processing steps take up 86% of a project's time
- Adoption is limited by knowledge and expertise

Background

- Machine Learning Pipelines
- Program Synthesis
- Monte Carlo Tree Search (MCTS)
- OpenML

Research question

How well does the Monte Carlo Tree Search algorithm perform in the context of program synthesis?

Methodology

1. Dataset

Retrieved using OpenML's API:

- 2 simple datasets (seeds, ilpd)
- 1 complex dataset (har)

2. Grammar

Properties:

- Context-free grammar
- Directed Acyclic Graphs
- Allows parallel processing steps

Operators from the scikit-learn library:

- Feature Preprocessing Operators (7)
- Feature Selection Operators (5)
- Supervised Classification Operators (5)

3. Search

Algorithm steps:

- Selection
- Expansion
- Simulation
- Backpropagation

4. Pipeline Generation and Evaluation

Optimization techniques:

- Subsampling
- Dynamic programming

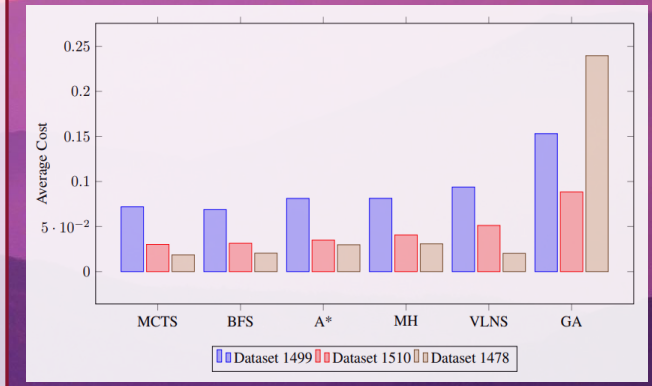
5. Performance Evaluation

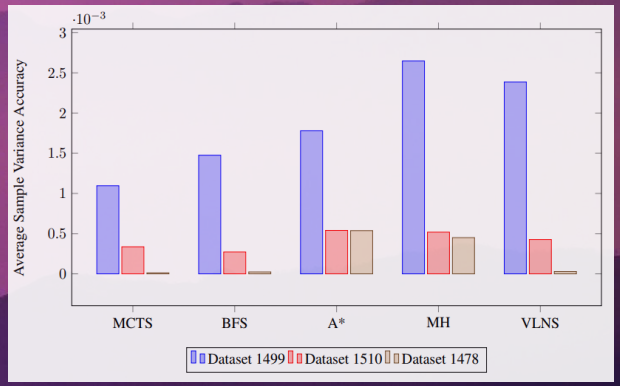
Evaluation metrics:

- Accuracy
- Cost
- Variance
- Time

Results

Metric	1499	1510	1478	Average
Accuracy	0.9281	0.9698	0.9814	0.9598
Cost	0.0719	0.0302	0.0186	0.0400
Variance	0.00110	0.00034	0.00001	0.00051
Execution time	10.9 s	166.0 s	168.7 s	115.2 s





Discussion

- Marginal improvement over BFS
- Constraints in time and resources
- Relatively simple datasets
- Algorithm's potential is underexplored

Conclusion

- Promising performance
- Full capabilities still uncertain
- Future research should focus on:
  - More challenging datasets
  - Algorithmic refinements

Author: Bastiaan Filius, B.L.Filius@student.tudelft.nl, +31 6 37437348, Supervisors: Sebastijan Dumančić, Tilman Hinnerichs