FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

BACHELOR THESIS

Gabriela Suchopárová

Evolutionary optimization of machine learning workflows

Department of Theoretical Computer Science and Mathematical Logic

Supervisor of the bachelor thesis:	Mgr. Roman Neruda, CSc.
Study programme:	Computer Science
Study branch:	General Computer Science

Prague 2019

I declare that I carried out this bachelor thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In .... date .... signature of the author

Dedication.

Title: Evolutionary optimization of machine learning workflows

Author: Gabriela Suchopárová

Department: Department of Theoretical Computer Science and Mathematical Logic

Supervisor: Mgr. Roman Neruda, CSc., Department of Theoretical Computer Science and Mathematical Logic

Abstract: Abstract.

Keywords: Machine learning Evolutionary computing Meta-learning Workflows

Contents

Introduction

1. Preliminaries

What we will talk about, theory

1.1 Machine learning

The field of machine learning encompasses a broad range of algorithms and statistical methods for data processing. In his book on machine learning, Flach provides the following general definition:

"Machine learning is the systematic study of algorithms and systems that improve their knowledge or performance with experience." (Flach [2012])

Here a detailed definition. Should be rewritten.

The exact meaning varies with different tasks. In some cases, knowledge is gained by processing labelled training data, in other cases it means comparing rewards of previous actions. The performance usually takes the form of some score — which can be accordance with some 'ground truth', i.e. comparison of the algorithm output with labelled testing data, or the success of an action.

With growing 'experience', the performance may increase as well. *Define overfitting, generalization error, how to avoid. Bias vs variance.*

1.1.1 Model ensembles

1.2 Metalearning

1.3 Evolutionary computing

Evolutionary computing is a heuristic method of optimization inspired by Charles Darwin's theory of *natural selection.* Darwin [1859] In a population, individuals with the best characteristics are most likely to reproduce, thus passing the traits to the offspring. As the evolution is repeated over several generations, the most advantageous traits predominate. This phenomenon is also called 'survival of the fittest'.

In an evolutionary algorithm, the goal is to find the "best" solution to the given problem. The term 'population' refers to a set of solutions encoded as chromosomes which represent the defining features of a particular solution. The 'natural selection' can be then understood as a stochastic search through the space of possible chromosome values. (Engelbrecht [2007])

A typical evolutionary algorithm

The advantage of this approach is that genetic algorithms perform multidirectional search, maintaining a population of potentially different solutions, which proves to be more robust than other directed search methods. (Michalewicz [1996], Mitchell [1997])

Algorithm 1: Evolutionary algorithm

```
Data: population size $n$, stopping condition $c$
Result: evolved individuals

$P(0) \longleftarrow$ population of size $n$
while $c$ is not met:
    for individual $ind$:
        compute fitness $f(ind)$
    for $i$ in range($n/2$):
        $i_1, i_2 \longleftarrow$ select two individuals
        if $p_{cx}$:
            perform crossover
        if $p_{mut}$:
            perform mutation
    $P(n+1) \longleftarrow$ select individuals from $P(n)$

return $P(c)$
```

1.4 Genetic programming

In this section, we present a subfield of evolutionary computing — the genetic programming — where the population is a set of computer programs. The aim of this technique is to evolve programs which provide a better solution to the given problem. There are various approaches in means of how to represent the individuals and what kind of genetic operators to use. The fitness is computed by running the program and comparing the result with the desired output. Poli et al. [2008]

1.4.1 Tree-based genetic programming

The individuals are most frequently represented in the form of *syntax trees.* Inner nodes of the tree are functions, whereas leaves are constants and variables. Both functions and constants are selected from a set of possible nodes which is provided as input to the algorithm.

1.5 Workflows

2. Related work

2.1 AutoML

2.2 TPOT

3. Our solution

Conclusion

Bibliography

Charles Darwin. *On the origin of species by means of natural selection, or, The preservation of favoured races in the struggle for life.* London, Murray, 1859.

Andries P. Engelbrecht. *Computational Intelligence: An Introduction.* Wiley Publishing, 2nd edition, 2007. ISBN 0470035617.

Peter Flach. *Machine Learning: The Art and Science of Algorithms That Make Sense of Data.* Cambridge University Press, New York, NY, USA, 2012. ISBN 1107422221, 9781107422223.

Zbigniew Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs (3rd Ed.).* Springer-Verlag, Berlin, Heidelberg, 1996. ISBN 3-540-60676-9.

Thomas M. Mitchell. *Machine Learning.* McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997. ISBN 0070428077, 9780070428072. URL `http://www.cs.cmu.edu/~tom/mlbook.html`.

Riccardo Poli, William B. Langdon, and Nicholas Freitag McPhee. *A Field Guide to Genetic Programming.* Lulu Enterprises, UK Ltd, 2008. ISBN 1409200736, 9781409200734.

List of Algorithms

List of Figures

List of Tables

List of Abbreviations

A. Attachments

A.1 First Attachment