Automatic Online Algorithm Selection

Hans Degroote HANS.DEGROOTE@KULEUVEN.BE
KU Leuven, Department of Computer Science, CODeS, iMinds-ITEC, Belgium

Patrick De Causmaecker PATRICK.DECAUSMAECKERL@KULEUVEN.BE
KU Leuven, Department of Computer Science, CODeS, iMinds-ITEC, Belgium

Keywords: Automatic Algorithm Selection, Reinforcement Learning, Contextual Multi-armed Bandits, Automatic Online Algorithm Selection

Abstract

In this abstract a reinforcement learning methodology for automatic online algorithm selection is introduced. Experiments confirm the usefulness of the methodology: using online data results in better performance. As in many online learning settings an exploration vs. exploitation trade-off, or synonymously a learning vs. earning trade-off, is incurred. The experiments show that a simple purely exploitative greedy strategy outperforms strategies explicitly performing exploration.

1. Introduction

The goal of this abstract is to outline a reinforcement learning methodology to extend the solution strategies for the problem of automatic algorithm selection from a purely offline setting to an online setting.

The field of algorithm selection is motivated by the observation that a unique best algorithm rarely exists for many problems. Which algorithm is best depends on the specific problem instance being solved. Such complementarities between different algorithms can be leveraged through algorithm portfolios (Huberman et al., 1997). Instead of using a single algorithm to solve a set of problems, several algorithms are combined in a portfolio and a method of selecting the most appropriate algorithm for each problem instance at hand is used. This selection process is called automatic algorithm selection.

Appearing in *Proceedings of Benelearn 2016*. Copyright 2016 by the author(s)/owner(s).

State of the art approaches for automatic offline algorithm selection use machine learning techniques to create a predictive model based on large amounts of training data. The predictive model predicts for each instance which algorithm is likely to be best. The model is created based on characteristics of the problem under consideration. These characteristics are called problem features. The idea is that their value should be correlated with how hard a problem is for a certain algorithm.

The observation motivating this research is that new performance data keeps being generated after the training phase every time the predictive model is used to predict the best algorithm for a new instance. This data is freely available yet not used by automatic offline algorithm selection methods. The main goal of this abstract is to answer the question: Can online performance data be used to improve the predictive model underlying automatic algorithm selection?.

2. Automatic Online Algorithm Selection Methodology

First automatic offline algorithm selection methodology will be discussed. Two broad classes of techniques for automatic offline algorithm selection can be identified. In the first fall classification-based techniques: the decision model directly predicts which algorithm will be best for an instance based on its features. No information about the actual quality of the algorithm is communicated.

The second class of automatic offline algorithm selection techniques consists of regression-based techniques. A regression model is created for each algorithm, predicting its performance in function of the problem features. The algorithm with the best predicted performance is selected to solve a new instance with.

Note that algorithm selection itself is a cost-sensitive classification problem (Bischl et al., 2012): the goal is to classify instances as belonging to the algorithm that best solves them and misclassification varies in cost (assigning an algorithm to its second best algorithm is less bad than assigning it to its worst). The distinction between classification and regression methods refers to how this classification problem is solved behind the scenes.

In offline approaches the decision models are kept fixed after the training phase. However, note that during the online phase performance data is generated every time a new instance is solved. This performance data consists of the performance of the selected algorithm on the new instance. The performance data of the other algorithms on the new instance is not available. Since this type of data can only be processed by the regression-based methods, the scope of the proposed methodology is limited to regression-based techniques.

The methodology for automatic online algorithm selection is the following. During the offline training phase an initial regression model is trained for each algorithm , using training data consisting of algorithm performance on instances described by feature values. During the online phase the algorithm to solve the first online instance with is selected based on the models created during the training phase. The model of the selected algorithm is retrained with the new datapoint. The performance of all other algorithms on the instance remains unknown. The algorithm to solve the second online instance with is selected based on the models created after having solved the first instance, thus one of the models has been updated to incorporate the performance information about the first online instance. Selection for the third online instance is influenced by the two previous etc. As more instances are solved, more datapoints are gathered and the models are expected to improve, which in turn is expected to result in better algorithm selection.

Adding additional data to the regression models is expected to result in better performance. To validate this hypothesis the standard non-learning strategy used by offline approaches is compared to a strategy that does process the online data. Experiments on data of ASLIB (Bischl et al., 2016), a standard benchmark within the automatic algorithm selection community, show that processing the additional online data does indeed improve performance.

An interesting challenge challenge faced in automatic online algorithm selection is finding a balance between learning a good predictive model and making good predictions during the online phase. Selecting a predicted non-best algorithm might be better in the long run because the information thus obtained results in a better model and more accurate predictions for future instances, but it negatively affects the expected performance on the current instance. This challenge is an example of the exploration vs. exploitation trade-off often faced in reinforcement learning.

Two basic strategies for solving the exploration vs. exploration trade-off that incorporate explicit exploration have been tested and compared to the purely exploitative approach. The exploration-incorporating strategies performed worse than the purely exploitative approach always greedily selecting the best algorithm. Furthermore, the exploration-incorporating strategies did not even learn better models. The immediate performance loss suffered by exploring was not compensated by any noticeable gain in learning.

3. Conclusions

A reinforcement learning methodology for automatic online algorithm selection has been introduced. Its scope is limited to automatic algorithm selection methods based on performance predictions for each individual algorithm. It has been shown experimentally that the method is capable of learning from online data and thereby improves on automatic offline algorithm selection methods.

It has been discussed that the exploration VS. exploitation trade-off should surface in automatic online algorithm selection, but the experiments suggest that the purely exploitative greedy strategy outperforms strategies that explicitly explore.

Acknowledgement

This abstract concerns work in collaboration with Bernd Bischl and Lars Kotthoff. The work presented in this abstract was submitted to the ITAT (Information technologies Applications and Theory) workshop and has been accepted.

Work supported by the Belgian Science Policy Office (BELSPO) in the Interuniversity Attraction Pole COMEX. (http://comex.ulb.ac.be).

The computational resources and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by the Research Foundation - Flanders (FWO) and the Flemish Government department EWI.

References

Bischl, B., Kerschke, P., Kotthoff, L., Lindauer, M., Malitsky, Y., Fréchette, A., Hoos, H., Hutter, F., Leyton-Brown, K., Tierney, K., et al. (2016). Aslib: A benchmark library for algorithm selection. *Artificial Intelligence*, *237*, 41–58.

Bischl, B., Mersmann, O., Trautmann, H., & Preuß, M. (2012). Algorithm selection based on exploratory landscape analysis and cost-sensitive learning. *Proceedings of the 14th annual conference on Genetic and evolutionary computation* (pp. 313–320).

Huberman, B. A., Lukose, R. M., & Hogg, T. (1997). An economics approach to hard computational problems. *Science*, *275*, 51–54.