Delft University of Technology

Alternating Maximization with Behavioral Cloning

Czechowski, A.T.; Oliehoek, F.A.

Publication date
2020

Document Version
Final published version

Published in
BNAIC/BeneLearn 2020

Citation (APA)
Czechowski, A. T., & Oliehoek, F. A. (2020). Alternating Maximization with Behavioral Cloning. In L. Cao, W. Kosters, & J. Lijffijt (Eds.), *BNAIC/BeneLearn 2020* (pp. 370-371). RU Leiden.
http://bnaic.liacs.leidenuniv.nl/wordpress/wp-content/uploads/papers/BNAICBENELEARN_2020_Final_paper_58.pdf

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

BNAIC BENELEARN 2020

Proceedings

November 19–20, 2020

Universiteit Leiden
The Netherlands

Editors: Lu Cao, Walter Kosters and Jefrey Lijffijt

Alternating Maximization with Behavioral Cloning

Aleksander Czechowski[1 [0000−0002−6054−9842]] and Frans A. Oliehoek[1 [0000−0003−4372−5055]]

Department of Intelligent Systems, Delft University of Technology, Delft, The Netherlands
{a.t.czechowski,f.a.oliehoek}@tudelft.nl

1 Introduction

The key difficulty of cooperative, decentralized planning lies in making accurate predictions about the behavior of one's teammates. In this paper we introduce a planning method of *Alternating maximization with Behavioural Cloning* (ABC) – a trainable online decentralized planning algorithm based on Monte Carlo Tree Search (MCTS), combined with models of teammates learned from previous episodic runs. Our algorithm relies on the idea of alternating maximization, where agents adapt their models one at a time in round-robin manner. Under the assumption of perfect policy cloning, and with a sufficient amount of Monte Carlo samples, successive iterations of our method are guaranteed to improve joint policies, and eventually converge.

2 The ABC method

Our planning algorithm is suitable for fully observable cooperative environments known as Multi-agent Markov Decision Processes (MMDPs). The setting is fully cooperative, and each agent is assumed to receive the same reward at each execution step of an episodic run. The planning is performed in a decentralized manner, and without communication between the agents. Each agent is equipped with an instance of the MCTS algorithm, a set of models of policies of its teammates, and a simulator of the environment. At each episodic step, each agent samples the simulator and teammate models to construct the tree of possible futures, estimate expected episodic rewards for individual actions, and choose the one which appears most beneficial.

Initially, agents are equipped with heuristic models of their teammates. They are assumed to act in a given environment repeatedly, for some large amount of episodic runs – either from simulation, or actual execution. Then, the agents use these experiences to learn to predict the actions of their colleagues. More specifically, every N episodic runs are grouped into one generation, and after each generation, the state-action episodic data, is used to train new agent models represented by convolutional neural networks; these are in turn provided to one of the agents, as the updated teammate models. At each generation only one agent updates its teammate models, which stabilizes training and, under certain assumptions on policy cloning, causes rewards to increase monotonically across the generations.

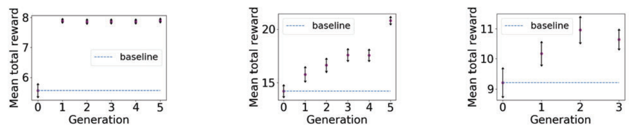

Fig. 1: Results from the factory floor experiment, in order of increasing difficulty. Left: two robots and preallocated tasks, middle: four robots and preallocated tasks, right: four robots and randomly appearing tasks. The baseline is the decentralized MCTS planning algorithm, introduced in [3].

3 Experiments

We test the efficiency of the algorithm by performing experiments in the spatial task allocation environment introduced in [2]. The domain consists of a gridworld-like planar map, where each position can be occupied by (cleaning) robots and tasks (e.g. litter). Each robot can perform either a movement action, which shifts the position of the robot accordingly, or a cleaning action, which removes one task at the current position. Attempted actions may succeed or not, according to predefined probabilities. Experiments show the effectiveness of the method, as an improvement across generations is observed, see Figure 1.

Acknowledgments

This project received funding from EPSRC First Grant EP/R001227/1, and the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 758824 —INFLUENCE).

References

1. Aleksander Czechowski and Frans A. Oliehoek. Decentralized MCTS via Learned Teammate Models Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, pages 81–88, 2020.
2. Daniel Claes, Philipp Robbel, Frans A. Oliehoek, Karl Tuyls, Daniel Hennes, and Wiebe Van der Hoek. Effective approximations for multi-robot coordination in spatially distributed tasks. Proceedings of the Fourteenth International Conference on Autonomous Agents and Multiagent Systems, pages 881–890, 2015.
3. Daniel Claes, Frans A. Oliehoek, Hendrik Baier, and Karl Tuyls. Decentralised online planning for multi-robot warehouse commissioning. Proceedings of the Sixteenth International Conference on Autonomous Agents and Multiagent Systems, pages 492–500, 2017.