Reinforcement Learning in POMDPs with Memoryless Options and Option-Observation Initiation Sets

Steckelmacher, Denis; Roijers, Diederik; Harutyunyan, Anna; Vrancx, Peter; Plisnier, Helene; Nowe, Ann

Reinforcement Learning in POMDPs with Memoryless Options and Option-Observation Initiation Sets

VRIJE UNIVERSITEIT BRUSSEL

D. Steckelmacher dsteckel@ai.vub.ac.be | D. M. Roijers droijers@vub.ac.be | A. Harutyunyan anna@harutyunyan.net | P. Vrancx peter@prowler.io | H. Plisnier hplisnie@vub.ac.be | A. Nowé anowe@vub.ac.be

- **Partial observability** is often solved using **memory**
- Complex hierarchical tasks are expressed using **Options**

→ We propose a **unified solution** to **Options** in **POMDPs**

Options can be combined with **recurrent neural networks**, but it is a **complicated** way to induce memory.

Option-Observation Initiation Sets (OOIs) **elegantly** allow **memoryless options** to encode **history** by making the set O_t of options available at time t depend on the current observation x_t and previous option ω_{t-1}

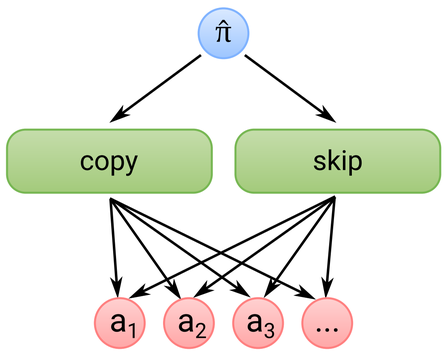
$$I_\omega \subset \Omega \times O \longrightarrow O_t = f(x_t, \omega_{t-1})$$

(I_ω: initiation set of ω; Ω: observations; O: options)

1 Background

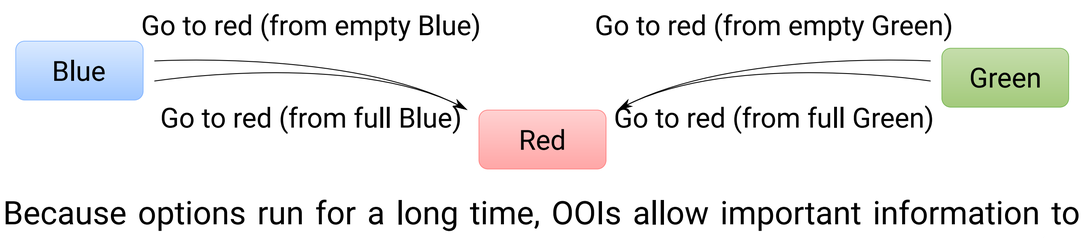
An agent, robot or program, learns how to perform sequences of actions in an environment so that its cumulative reward is maximized. At each time-step, the agent observes $x_t \in \Omega$, selects action a_t and obtains a reward r_t +1.

An option [1] is like an sub-policy that executes for several time-steps. The top-level policy selects an option, that selects actions until it terminates. The top-level policy then selects another option. An option $\omega \in O$ can be selected at time t only if x_t is in its initiation set I_ω. Therefore, the **initiation sets constrain which options are available in which situations**.



2 OOIs Lead to an Implicit Memory

If several options share the same policy, but are activated in different situations, which one terminates at a given time tells the agent **what was observed back when it was selected**. OOIs allow the agent to make use of this information, by selecting another option based on which one terminated.



Because options run for a long time, OOIs allow important information to be remembered over large time horizons, **without information decay** (as happens with recurrent neural networks [2]). Combined with the intuitiveness of Options with OOIs, this makes our approach perfectly suited for safety-critical or robotic RL.

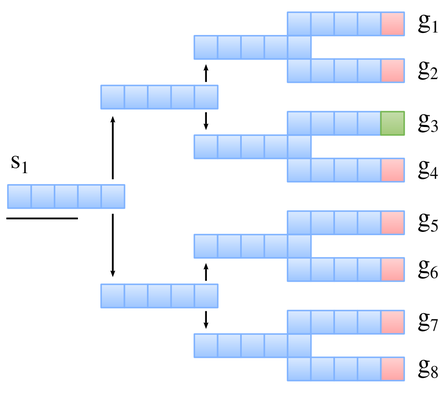
Options with OOIs, despite their simplicity, combine all the advantages of memory bits and finite-state controllers, while being compatible with all the current reinforcement learning algorithms.

3 Experiments

Terminals: A robot has to gather objects, one by one, from two terminals, and carry them to a central location. After some time, a terminal may become empty, at which point the other one is refilled. A wall between the terminals prevents the robot from going from one to the other directly, it has to pass by the central location. Whether a terminal is full or empty cannot be observed from the central location, so has to be remembered.

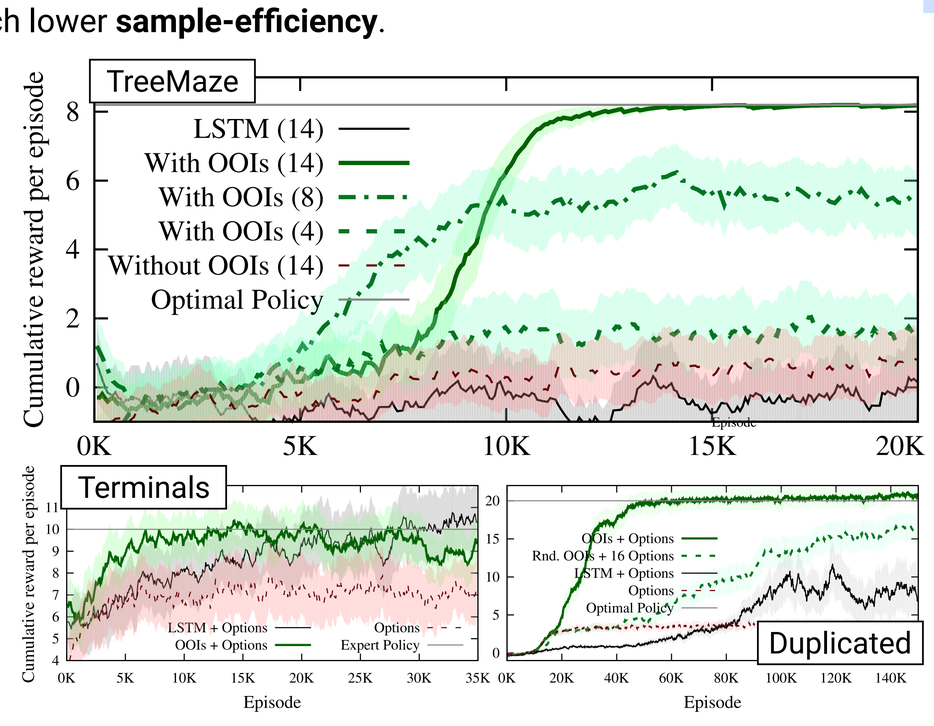
DuplicatedInput: The agent has to copy characters one by one from an input tape to an output tape. B's and D's always appear in pairs and must be deduplicated. The agent does not observe its position on the input tape.

TreeMaze: The agent has to navigate to one of the 8 leaves of a tree. The desired leaf is selected at random for each episode. The agent observes which leaf to go to during 3 time-steps, then has to remember that information until it reaches the goal.



4 OOIs Outperform LSTM over Options

A recurrent neural network, containing 20 LSTM units, is also evaluated. The LSTM agent has access to the same options as the OOIs agent, but has standard initiation sets instead of ours. It also learns the task but has much lower **sample-efficiency**.



References

[1]: Sutton, Precup, Sigh, *Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning*, Artificial Intelligence, 1999

[2]: Bakker. *Reinforcement Learning with Long Short-Term Memory*. NIPS, 2001.

Acknowledgments

The first author is "Aspirant" at the Research Foundation - Flanders (FWO), grant number 1129317N. The second author is "Postdoctoral Fellow" at the FWO, grant number 12J0617N.