Hierarchical Reinforcement Learning Based on Subgoal Discovery and Subpolicy Specialization

Bram Bakker

IAS, Informatics Institute, University of Amsterdam
Kruislaan 403, 1098 SJ Amsterdam
bram@science.uva.nl

Abstract

A method is described for hierarchical reinforcement learning. High-level policies automatically discover subgoals; low-level policies learn to specialize for different subgoals. Subgoals are represented as desired abstract observations which cluster raw input data. Experiments shows that this method outperforms several flat methods. The full paper can be found in [1].

Introduction. The promise of scaling up reinforcement learning (RL) through *hierarchical* RL (HRL) is widely acknowledged. The idea is that low-level policies, which emit the actual "primitive" actions at a fast timescale, solve only parts of the overall task. Higher-level policies work on a slower timescale, solving the overall task by sequentially invoking lower-level policies, considering only a few abstract high-level observations and actions (macro-actions, options). Thus each level's search space is reduced, temporal credit assignment is facilitated, and low-level policies become easily reusable within the task and in different tasks.

In most previous HRL work the hierarchical structure was prewired by a designer. To minimize the latter's responsibility, however, we would like to learn the hierarchical structure as well. This work [1] presents a step in that direction. We propose the HASSLE (Hierarchical Assignment of Subgoals to Subpolicies LEarning) algorithm, in which high-level policies automatically discover subgoals, and low-level policies learn to specialize for different subgoals. In what follows the intuition behind HASSLE is explained, [1] contains a more formal description.

The HASSLE algorithm. Both high-level and low-level policies learn using essentially standard value function-based reinforcement learning algorithms. The high-level value function covers the complete state space at a coarse level. It is updated based on "external" rewards, received through interaction with the environment. Low-level value functions cover only parts of the overall state space, at a fine-grained level. They are updated based on "internal" rewards provided by the high-level policy.

Each action of the high-level policy is the selection of a subgoal in response to a new high-level observation. A standard value function-based RL algorithm is used to learn this mapping, using external reward signals received when goals are reached as defined by the task. High-level observations correspond to clusters

of low-level observations (e.g. obtained through an unsupervised clustering algorithm). Both the current high-level observation and the subgoal come from this set of high-level observations; thus, the subgoal is the high-level observation that the high-level policy "wants to see next". The use of high-level observations (state abstraction) is one difference with most existing HRL algorithms.

It is the job of the low-level policies to reach the subgoal selected by the high-level policy. There is a limited, fixed set of low-level policies. None of them is initially associated with any of the possible subgoals. However, in addition to a standard state-action value function, every low-level policy contains a table of *C-values* (initially set to 0). Each C-value represents the "Capability" of this low-level policy to reach one of the subgoals. Based on the C-values of all low-level policies for the current subgoal, one low-level policy is selected (stochastically). This one then attempts to reach the current subgoal. If it reaches the subgoal, it receives a positive internal reward, which is used to update the C-value for this subgoal, making future selection more likely. If it does not reach the subgoal, it receives zero internal reward, the C-value is updated accordingly, and future selection becomes less likely. In this way, a low-level policy may learn that it can reach subgoal A but not subgoal B. This realizes *specialization*. But it may also learn that its capability encompasses multiple subgoals. This realizes *generalization*. Thus, low-level policies will specialize when they have to, but generalize when they can. This idea of low-level policies learning to specialize based on learning additional values is another important difference with most existing HRL algorithms.

The internal reward provided by the high-level policy to the active low-level policy is also used to update the latter's state-action value function, again using a standard RL algorithm. Function approximators are used to represent these state-action value functions, such that each low-level policy can learn to focus on those parts of the low-level observation space which are relevant to its specialization.

Experiments. Experiments were done in two large "office navigation" MDPs, one of which stochastic, in order to test HASSLE and compare it to standard, "flat" RL methods. Results indicate that, in the example MDPs at least, HASSLE works well and learns significantly faster than several flat RL methods, if the latter worked at all. Low-level policies develop meaningful specializations: for instance, in the office navigation task one low-level policy is used to exit rooms (and this is all it can do), and another low-level policy is only used to navigate through corridors.

References

[1] B. Bakker and J. Schmidhuber. Hierarchical reinforcement learning based on subgoal discovery and subpolicy specialization. In F. Groen, N. Amato, A. Bonarini, E. Yoshida, and B. Kröse, editors, *Proceedings of the 8-th Conference on Intelligent Autonomous Systems, IAS-8, Amsterdam, The Netherlands*, pages 438–445, 2004.