Learning Relative Return Policies With Upside-Down Reinforcement Learning

Dylan R. Ashley [1,2,3]

Kai Arulkumaran [4,5]

Jürgen Schmidhuber [1,2,3,6,7]

Rupesh Kumar Srivastava [7]

1 The Swiss AI Lab IDSIA

2 Università della Svizzera italiana

3 Scuola universitaria professionale della Svizzera italiana

4 ARAYA Inc.

5 Imperial College London

6 AI Initiative, King Abdullah University of Science and Technology

7 NNAISENSE

Abstract

Lately, there has been a resurgence of interest in using supervised learning to solve reinforcement learning problems. Recent work in this area has largely focused on learning command-conditioned policies. We investigate the potential of one such method—upside-down reinforcement learning—to work with commands that specify a desired relationship between some scalar value and the observed return. We show that upside-down reinforcement learning can learn to carry out such commands online in a tabular bandit setting and in CartPole with non-linear function approximation. By doing so, we demonstrate the power of this family of methods and open the way for their practical use under more complicated command structures.

arXiv

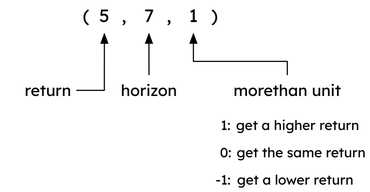
2202.12742

1

1: $\pi \leftarrow$ initial policy
2: **while** not done **do**
3: pick a command C_0 of the form $(desire, horizon)$
4: generate experience by selecting actions according to π and C_0
5: **for** $S_t, A_t, (D_t, H_t), G_t \in$ new experience **do**
6: train π using input $(S_t, (G_t, H_t))$ and desired output A_t
7: **end for**
8: **end while**

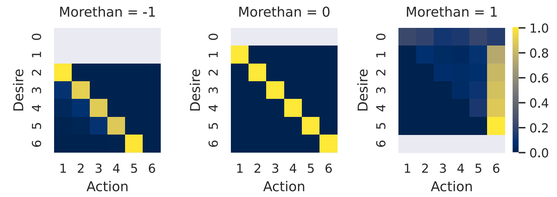
While the formal UDRL paradigm is more general than the above simple version, the underlying principles are the same. The basic idea of UDRL is to have one agent learning command-conditioned policies and a second agent issuing the first agent increasingly challenging commands. The first agent is then trained with the hindsight method.

2



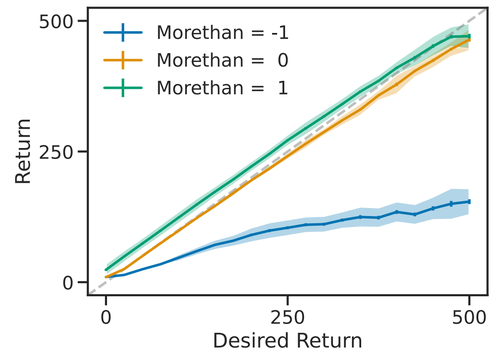
The alternate command structure for UDRL we explored here augments the basic command structure with a morethan unit that allows specifying goals in a relative way.

3



The mean action probabilities of the learned command-conditioned policy in the bandit setting. Note how the probability densities seem to cluster around the highest-valued valid action.

4



The desired return versus the observed return under the learned command-conditioned policy in the CartPole setting. Standard deviation is shown with shading. Note how the observed returns center around the lower-end of the valid returns.