Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Learning Options in Multiobjective Reinforcement Learning

Rodrigo Cesar Bonini, Felipe Leno da Silva, Anna Helena Reali Costa*
Escola Politécnica da Universidade de São Paulo, São Paulo, Brazil
Av. Prof. Luciano Gualberto, n.158, Engenharia Elétrica, Cidade Universitária, São Paulo, CEP: 05508-970.
{rodrigo_cesarb,f.leno,anna.reali}@usp.br

Abstract

Reinforcement Learning (RL) is a successful technique to train autonomous agents. However, the classical RL methods take a long time to learn how to solve tasks. Option-based solutions can be used to accelerate learning and transfer learned behaviors across tasks by encapsulating a partial policy into an action. However, the literature report only single-agent and single-objective option-based methods, but many RL tasks, especially real-world problems, are better described through multiple objectives. We here propose a method to learn options in Multiobjective Reinforcement Learning domains in order to accelerate learning and reuse knowledge across tasks. Our initial experiments in the *Goldmine* Domain show that our proposal learn useful options that accelerate learning in multiobjective domains. Our next steps are to use the learned options to transfer knowledge across tasks and evaluate this method with stochastic policies.

Introduction

The RL Framework (Sutton and Barto 1998) allows autonomous agents to learn with interactions in the environment. Many sequential decision problems are modeled by a Markov Decision Process (MDP)(Sutton and Barto 1998), and RL is an extensively used solution for MDPs through environment interactions. An MDP is described by the tuple $\langle S, A, T, R\rangle$, where S is the set of environment states, A is the set of available actions, T is the transition function, and R is the reward function (the agent does not know T and R). The agent goal is to learn an optimal policy π^*, that maps the best action for each possible state.

Although RL has been succesfully applied in many problems, its classical approaches learns very slowly. Learning π^* may take too many time, because RL classical approach needs many steps to explore the state-action space.

Further, many RL tasks cannot be easily described by a single reward function, and are rather described by multiple reward functions. Many tasks, as the Multiobjective Reinforcement Learning (MORL)(Van Moffaert, Drugan, and Nowé 2013) ones, are better described through multiple reward functions $\{R_1, \ldots, R_i\}$, as they require the maximization of multiple and possibly conflicting objectives. MORL methods solve such tasks by balancing all objectives as well as possible, but scalability issues are further intensified and the learning process may becomes even slower. Hence, as domains become progressively complex, scalability gains more importance for RL methods.

The Options Framework (Sutton, Precup, and Singh 1999) was proposed to alleviate scalability issues. Options are high-level actions that encapsulate a partial policy, often representing the solution of a subtask. Including well-built options into the action space accelerate learning. While in the original Options Framework each option was provided by a domain expert, later works like the *PolicyBlocks Algorithm* (Pickett and Barto 2002), propose autonomous option-discovery methods through the evaluation of previous task solutions. Furthermore, learned options are reported to accelerate learning even when transferred across similar tasks.

The *PolicyBlocks Algorithm* autonomously learn useful options by analyzing previous task solutions(optimal policies) and finding commonly occuring partial policies, named options. However, as well as in other option-discovering solutions, only single-objective domains are take into account.

Besides that, although MORL algorithms aim at solving multiple objective problems, there is no MORL approach from option-discovery methods until now and even the effect of human-specified options in MORL is not regarded in the literature.

Therefore, we argue that MORL algorithms can also benefit from option-based solutions to accelerate learning. We here propose a method, hereafter called Multiobjective Options (MO-Opt), to learn options in multiobjective domains.

Our initial experiments show that our proposal can autonomously learn useful options that accelerate learning in multiobjective domains. The next step is to evaluate our option-discovery algorithm when transferring options across different domains.

Proposal

We here introduce MO-Opt, an approach based in the Options Framework, for option-discovery in multiobjective domains. The main idea of MO-Opt is to learn options for each of the objectives separately (for which the *PolicyBlocks* al-

*We are grateful for the support from the CEST Group, CNPq (grant 311608/2014-0), and São Paulo Research Foundation (FAPESP), grant 2015/16310-4.

gorithm, for example, can be used) and apply the learned options to the multiobjective problem. The learned options are intended to optimize a single objective, but may maximize multiple objectives for some situations, or guide the agent towards trajectories that maximize one objective without hampering the others. The RL algorithm is able to identify when each option is useful, and we argue that these options can accelerate the learning process, guiding the agent's exploration to learn a given task faster.

Algorithm 1 fully describes our proposal. Firstly we initialize a set of options Π. Then, we learn a set of suboptimal policies L_i for each objective $i \in O$ by using a standard RL algorithm, where O is the set of objectives. After that, we use the standard *PolicyBlocks Algorithm* to learn a set of options for each objective separately (including them in Π), but any other options-discovery algorithm could also be used. Finally, the Multiobjective learning algorithm is executed using $\Pi \cup A$, which is expected to accelerate learning.

Algorithm 1 Learning Options with MO-Opt

```
Π ← ∅
for each objective i ∈ O do
    for H episodes do
        learn a set of optimal policies L_i for i
        Π ← Π ∪ PolicyBlocks(L_i)
    end for
end for
run MORL using Π ∪ A
```

Experimental Evaluation

In order to evaluate our proposal, we choose a 18x18 *Goldmine* Domain, in which the agent must optimize two (possibly) conflicting objectives. A certain number of *gold pieces* are spread in the environment, and the agent has a goal position to be reached. The first objective is to arrive in the goal position as fast as possible, and the second objective is to collect all *gold pieces*. However, *gold pieces* are often outside the optimal path towards the goal position, then the agent must balance the two objectives and reason over the objective to be pursued at a certain moment.

We placed 4 *golds* in the map, and the action set is $A = \{north, south, east, west\}$. A *gold piece* is collected when the agent is in the same position of it, which results in a reward of +1 in the second reward function. A reward of +1 is awarded in the first reward function when the agent achieves the goal position, and then the episode ends. Otherwise, the reward is 0 for both objectives.

Firstly, we performed the Q-learning algorithm for each objective for 100 episodes, providing 5 optimal policies to *PolicyBlocks*, that chooses the 3 best scored options for each objective. Then, we executed 100 learning episodes using the *Scalarized Q-Learning algorithm* (Silva and Costa 2015) both without options and using MO-Opt, in order to evaluate the relative effectiveness of the learned options to the learning process. The scalarized algorithm was configured with the weights w_1 = 0.75 and w_2 = 0.25, i.e., the reward functions are combined using a linear combination with those weights. We set the learning rate α = 0.2 and the discount rate γ = 0.9 in our experiments. The observed performances are averages over 100 executions of this procedure.

MO-Opt learns faster at the beginning of the learning processes, achieving an average reward of 0.16 after roughly 30 learning episodes. The standard Q-Learning without options achieved similar results only after approximately 60 learning episodes. This difference between the average cumulative reward indicated that MO-Opt provides a better jump-start to MORL domains than standard RL techniques.

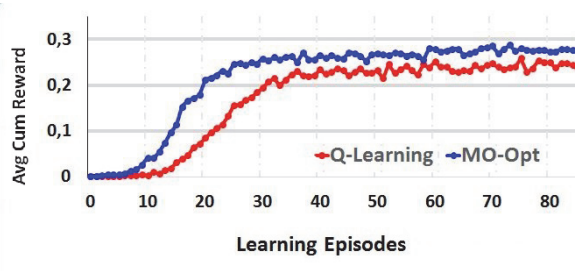

Figure 1: The average reward (linear combination of the two objectives) for 80 episodes during the learning process.

Conclusion and Further Works

The main conclusion of this experiment is that the options-discovering methods outperformed regular learning even though in MORL domains which the agents had no previous knowledge. Our experiments in *Goldmine* Domain showed that our approach is promising for accelerating learning in multiobjective domains. The learning process was accelerated as expected, because like in human learning processes, is faster to learn when having previous knowledge. The next step is to transfer the learned Options across different domains and evaluate if the learning process of related yet different tasks is accelerated by our proposal and evaluate this method with stochastic policies.

References

Pickett, M., and Barto, A. G. 2002. Policyblocks: An algorithm for creating useful macro-actions in reinforcement learning. In *ICML*, 506–513.

Silva, F. L., and Costa, A. H. R. 2015. Multi-objective reinforcement learning through reward weighting. In *TRI 2015, joint with IJCAI*, volume 1, 25 – 36.

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1st edition.

Sutton, R. S.; Precup, D.; and Singh, S. 1999. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence* 112(1-2):181–211.

Van Moffaert, K.; Drugan, M. M.; and Nowé, A. 2013. Scalarized multi-objective reinforcement learning: Novel design techniques. In *ADPRL*, 191–199. IEEE.