Transferring Probabilistic Options in Reinforcement Learning

Rodrigo Cesar Bonini, Felipe Leno da Silva, Ruben Glatt and Anna Helena Reali Costa
Escola Politécnica da Universidade de São Paulo, São Paulo, Brazil
{rodrigo_cesarb, f.leno, ruben.glatt, anna.reali}@usp.br

ABSTRACT

Option-based solutions can be used to accelerate learning and transfer learned behaviors across tasks by encapsulating a partial policy. However, commonly these options are specific for a single-task, and may ignore some good and alternative solutions when this knowledge is transferred and reused in another task. Furthermore, these solutions may provide bad partial policies to the agent, making the learning process worse than without the use of options. We here propose a multi-task method to combine learned options into a probabilistic one in order to enable better choices to the agent, so it does not get stuck in certain regions or follows bad partial solutions. Our experiments in the *Gridworld* Domain show that our proposal learns useful options that accelerate learning while also providing alternative decisions to the agent.

Keywords

Reinforcement Learning, Options, Transfer Learning, Probabilistic Policies

1. INTRODUCTION

The RL Framework [9] allows autonomous agents to learn through interactions with an environment. Many sequential decision problems are modeled as a Markov Decision Process (MDP)[9] which can be solved by Reinforcement Learning (RL) algorithms.

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. In RL the agent does not know T and R, and the 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 [12, 6, 5], its classical approaches learn very slowly and learning the optimal policy π^* may take too long because they need many steps to explore the whole state-action space.

On the one hand, Options Framework [10], which can be easily incorporated into a variety of different RL algorithms, offers a way to propose high-level actions that encapsulate sequences of actions performed by agents, accelerating their learning process. An option may solve a sub-goal in a RL problem and may contain a subset of optimal actions for certain states [13]. For instance, an option in a indoor navigation domain could portray some or all the actions needed to move towards a door, unlock it, and open it.

On the other hand, Transfer Learning (TL) solutions[11, 1, 2, 8] allow to reuse knowledge acquired in previous tasks, generalizing and transferring knowledge between tasks or agents, thus accelerating learning in RL domains. A possible way to transfer knowledge across tasks is to reused learned policies in new tasks [3, 4]. This approach works like human learning, where previous and partial knowledge can be used to accelerate the learning of new human tasks.

The reason for this is that options capture only a fixed structure of a task.

A framework proposed in [4] explores the idea that generalization from closely related, but solved problems can produce policies that provide good decisions in many states of a new unsolved task, because it avoids the possibility of a bad performance of a known policy in a new task, indicating that abstract and non-deterministic policies can offer an effective guidance to the agents. These benefits happen on the exploration strategy, so the algorithm converges much faster than without transfer of knowledge.

Another approach was proposed in [3] to probabilistically reuse a set of past deterministic policies that solve different tasks within the same domain. It features an autonomously growing library of policies, which stores the most different policies in order to identify core policies of a domain that give the greatest advantage when solving a new task.

Here we argue that the use of options is better than policies since they represent parts of the solution, encapsulating behaviors necessary to solve many related problems. Moreover, we argue that probabilistic options are important knowledge to be transferred from one task to a new task, in order to assist the agent in its learning process.

Our initial experiments show that our proposal can autonomously learn useful options that improve learning for similar tasks.

2. PROPOSAL

We here introduce a method, hereafter called *Probabilistic Combined Options* (PCO) to combine learned options from different tasks in order to provide a good initialization set of actions according to previously learned options, without taking away the possibility of the agent visiting states that still unexplored.

Appears in: *Proceedings of the 1st Workshop on Transfer in Reinforcement Learning (TiRL) at the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017), A. Costa, D. Precup, M. Veloso, M. Taylor (chairs), May 8–9, 2017, São Paulo, Brazil.*

The idea of PCO is to learn options for each source task independently (for which the *PolicyBlocks* algorithm [7], for example, can be used) combine them and apply the combined probabilistic option in the target task.

The learned options are intended to optimize the learning process in the target task, guiding the agent towards probably good trajectories, while allowing it to explore other new states.

Our proposal is fully described by Algorithm 1. Firstly it initializes a set of options Φ. Then, it learns a set of suboptimal policies L_ψ for each source task $\psi \in \Psi$ by using a standard RL algorithm, where Ψ is the set of source tasks. After that, it uses the standard *PolicyBlocks Algorithm* (other options-discovery algorithms may be used, but were not evaluated here) to learn a set of options for each source task independently (including them in Φ).

Finally, the options are combined into a probabilistic option Ω and the learning process is then executed in the target task with the resulting option Ω as another choice in the set of actions.

The options are combined in a way that Ω selects actions with a probability according to the number of times that they appear in the previous options, where the more the action appears, the greater its probability. Here, Ω executes each action with a probability given according to:

$$p(a|s)_\Omega = \frac{|\Phi(a|s)| + 1}{|\Phi_s| + |A|}, \quad (1)$$

where $p(a|s)_\Omega$ is the probability of action a being chosen by Ω in state s, $|\Phi(a|s)|$ is the number of options in Φ that select a in state s, and $|\Phi_s|$ is the number of options that are defined for state s. Here, the more an action appears in a certain state inside the discovered options, the greater is its chance of being chosen by the agent.

Algorithm 1 PCO

1: $\Phi \leftarrow \emptyset$
2: **for each** source task $\psi \in \Psi$ **do**
3: **for** H episodes **do**
4: learn a set of policies L_ψ for ψ
5: $\Phi \leftarrow \Phi \cup PolicyBlocks(L_i)$
6: **end for**
7: **end for**
8: $\Omega \leftarrow combine(\Phi)$
9: run learning in target task using $\Omega \cup A$

3. EXPERIMENTAL EVALUATION

To evaluate our proposal, all the experiments were performed in a 11x11 *Gridworld Domain*, in which the agent starts in a random non-terminal state and must perform 6 source tasks (one task at a time) with a different goal position to be reached in each of them.

Firstly, the agent has to learn how to achieve the goal position as fast as possible independently in the 6 source tasks and the options for each task are discovered and stored.

We performed the Q-learning algorithm for 1000 episodes, providing 5 policies to *PolicyBlocks* extracts 3 options for each task. After that, those options Ω are combined and the resulting option is evaluated in 6 target (different) tasks.

The action set available after learning the options is $A = \{north, south, east, west\} \cup \Omega$. Episodes ends when the agent achieves the goal state, resulting in a reward of +1 discounted by $\gamma = 0.9$ and otherwise, the reward is 0 for any step. We also adopted the learning rate $\alpha = 0.2$.

In order to evaluate the relative effectiveness of the probabilistic learned options, we executed 1000 learning episodes using the Vanilla Q-Learning algorithm without options and 1000 learning episodes of our approach, PCO.

Figure 1 shows the average discounted reward in 1000 repetitions of the experiment. PCO outperformed the Regular Q-Learning, learning faster at the beginning of the learning processes, achieving an average reward of 0.30 after only 10 learning episodes.

The standard Q-Learning without options was not able to achieve similar results even after 1000 learning episodes.

This difference between the average cumulative reward indicates that PCO provides a speed-up in the learning process, greatly improving the performance in the initial learning episodes, while also providing alternative decisions to the agent.

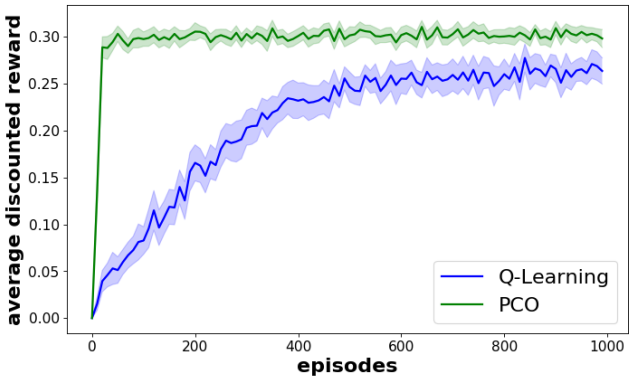

Figure 1: The average reward for 1000 episodes during the learning process.

4. CONCLUSION AND FURTHER WORKS

The main contribution of this work is that the probabilistic options combined provide to the agents good alternatives based on previous knowledge.

Our experiments in the *Gridworld* Domain show that our approach is promising both for accelerating learning and guiding the agents to good solutions in RL domains.

In the future, we intend to evaluate the options before combine them, compare probabilistic options with non-deterministic policies [4], portable transfer options [13], and multiobjective options [2]. We also intend to evaluate the approach in more complex domains and compare it to other option-discovery methods. Finally, we plan to adapt our method to enable the transfer of learned options across different domains.

Acknowledgements

We are grateful for the support from the CEST Group, CNPq (grant 311608/2014-0), São Paulo Research Foundation (FAPESP), grants 2015/16310-4 and 2016/21047-3, and CAPES.

REFERENCES

[1] R. A. Bianchi, L. A. C. Jr., P. E. Santos, J. P. Matsuura, and R. L. de Mantaras. Transferring knowledge as heuristics in reinforcement learning: A case-based approach. *Artificial Intelligence*, 226:102 – 121, 2015.

[2] R. C. Bonini, F. L. da Silva, E. Spina, and A. H. R. Costa. Using options to accelerate learning of new tasks according to human preferences. In *Human-Collaborative Learning Workshop (AAAI-17)*, page (Accepted Paper), 2017.

[3] F. Fernández and M. Veloso. Probabilistic policy reuse in a reinforcement learning agent. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, AAMAS '06, pages 720–727, New York, NY, USA, 2006. ACM.

[4] M. L. Koga, V. F. da Silva, and A. H. R. Costa. Stochastic Abstract Policies: Generalizing Knowledge to Improve Reinforcement Learning. *IEEE Transactions on Cybernetics*, 45(1):77–88, 2015.

[5] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[6] A. Y. Ng, A. Coates, M. Diel, V. Ganapathi, J. Schulte, B. Tse, E. Berger, and E. Liang. Autonomous inverted helicopter flight via reinforcement learning. In *Experimental Robotics IX*, pages 363–372. Springer, 2006.

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

[8] V. Soni and S. Singh. Using homomorphisms to transfer options across continuous reinforcement learning domains. In *AAAI*, volume 6, pages 494–499, 2006.

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

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

[11] M. E. Taylor and P. Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10:1633–1685, 2009.

[12] G. Tesauro. *TD-Gammon: A Self-Teaching Backgammon Program*, pages 267–285. Springer US, Boston, MA, 1995.

[13] N. Topin and J. MacGlashan. Portable option discovery for automated learning transfer in object-oriented markov decision processes. In *Proceedings of the 24th International Conference on Artificial Intelligence*. AAAI Press, 2015.