Providing Uncertainty-Based Advice for Deep Reinforcement Learning Agents (Student Abstract)

Felipe Leno Da Silva,[1*] **Pablo Hernandez-Leal,**[2] **Bilal Kartal,**[2] **Matthew E. Taylor**[2]
[1]University of São Paulo, Brazil
[2]Borealis AI, Canada
f.leno@usp.br, {pablo.hernandez, bilal.kartal, matthew.taylor}@borealisai.com

Abstract

The sample-complexity of Reinforcement Learning (RL) techniques still represents a challenge for scaling up RL to unsolved domains. One way to alleviate this problem is to leverage samples from the policy of a demonstrator to learn faster. However, advice is normally limited, hence advice should ideally be directed to states where the agent is uncertain on the best action to be applied. In this work, we propose *Requesting Confidence-Moderated Policy advice* (RCMP), an action-advising framework where the agent asks for advice when its uncertainty is high. We describe a technique to estimate the agent uncertainty with minor modifications in standard value-based RL methods. RCMP is shown to perform better than several baselines in the Atari Pong domain.

Introduction

Reinforcement Learning (RL) requires exploring the environment for gathering samples, which makes the learning process potentially costly or dangerous for many real-world applications. For this reason, many of the recent investigations in the area aim at reducing the amount of required samples during learning.When a high performance policy is available, the learning agent might leverage samples of it (hereafter referred as *advice*) to reduce the need of random exploration at the beginning of learning. However, the amount of advice is often limited by human availability or communication costs, hence the learning agent must make effective use of the available samples, while taking into account the demonstrator availability. In the literature, advice is usually given at a prefixed frequency or based on heuristics unrelated to the agent uncertainty. This means that the advice is not aimed to disambiguate situations in which the agent has high uncertainty and the demonstrator is likely to give advice when the agent does not need it. We propose *Requesting Confidence-Moderated Policy advice* (RCMP), an algorithm to selectively give advice to a learning agent in situations where its uncertainty is high. When the agent has a low uncertainty, this indicates that its value estimate for the current state is close to convergence, and for this reason advice might be saved for more useful situations. We also describe an approach to estimate uncertainty for model-free RL algorithms. This measure is used by RCMP to define if advice should be given in the current state. Our approach consists in extracting multiple estimates of Q-values from a single network, similarly as done in Bootstrapped DQN (Osband et al. 2016). The variance between those estimates is then used as a metric of the uncertainty, used to define when advice is expected to be useful. We show in our empirical evaluation that our method is able to make efficient use of advice, performing better than related baselines.

[*]This work was completed while an intern at Borealis AI.

Background

RL enables the solution of *Markov Decision Processes* (MDP). An MDP is described by a tuple $\langle S, A, T, R\rangle$. S is the set of states in the system, A is the set of actions available to the agent, T is the state transition function, and R is the reward function. The goal of the agent is learning a policy $\pi : S \rightarrow A$ that dictates the action to be applied in each possible state, where the optimal policy π^* maximizes the expected reward achieved. Commonly, RL algorithms aim at learning a state-action value function (or Q-function) that approximates the expected return of applying each action in a particular state. The optimal Q-function is $Q^*(s,a) = E\left[\sum_{i=0}^{\infty}\gamma^i r_i\right]$, where r_i is the reward received after i steps from using action $a = \pi^*(s)$ on state s, and γ is a discount factor. Q can be used to extract a policy $\pi(s) = \arg\max_{a\in A} Q(s,a)$, where using Q^* results in π^*. Deep Q-Network (DQN) (Mnih et al. 2015) leverages *Deep Neural Networks* to learn Q-functions. Although effective, DQN and other similar algorithms suffer from high-sample complexity. However, for many tasks, a competent policy can be accessed before starting the training process, e.g., because a human can provide some examples of how to solve the task. *Action Advice* (Silva and Costa 2019), a way of leveraging those policies, consists of receiving action advice for a single state where it is expected to be useful. One of most used metrics for defining when to give advice is the *importance advising*: $I(s) = \max_{a\in A} Q_{\mathcal{D}(s,a)} - \min_{a\in A} Q_{\mathcal{D}(s,a)}$ (Torrey and Taylor 2013; Amir et al. 2016), where $Q_{\mathcal{D}}$ is the Q-table of the demonstrator. This method has two drawbacks addressed by our proposal: (i) the advising-function

does not consider the learning agent policy, and advice is possibly given in states where the policy is already good; (ii) the demonstrator has to observe the agent constantly.

Uncertainty-based Advice

We propose the *Requesting Confidence-Moderated Policy advice* (RCMP) algorithm aiming at leveraging action advice based on the agent uncertainty to accelerate learning. We assume that a demonstrator $\pi_{\mathcal{D}} : S \times A \rightarrow [0, 1]$ is available to the agent and can be queried to give action suggestions. Before taking any action in the environment, RCMP queries the agent uncertainty estimate $\mu(s)$. In case the uncertainty is high, the agent asks for advice and executes the action suggested by $\pi_{\mathcal{D}}(s)$. Otherwise, the usual exploration is strategy is executed based on the current policy π. Vanilla Q-function based algorithms are unable to provide an estimate of the agent's uncertainty on its value predictions. Therefore, we propose a way to measure the uncertainty with a small modification in standard Q-based algorithms. Consider the illustration of a modified DQN network in Figure 1. We propose to add as a last layer of the value function neural network multiple *heads* estimating separately expected values for each action, as done in Bootstrapped DQN 2016. Due to the aleatoric nature of the exploration and network initialization, each head will initially output a different estimate of the action values, but the variance between them will be progressively reduced during training as they get closer to convergence. Hence, we use the variance between the predictions as uncertainty measure: $\mu(s) = \frac{\sum_{\forall a \in A} var(\boldsymbol{Q}(s,a))}{|A|}$, where $\boldsymbol{Q}(s,a) = \begin{bmatrix} Q_1(s,a) \\ \vdots \\ Q_h(s,a) \end{bmatrix}$, $Q_i(s,a)$ is the Q-value given by head i for state s and action a, var is the variance, and h is the chosen number of heads. The final value prediction (used, for example, for extracting a policy from the value function) is the average of the predictions given by each head: $\hat{Q}(s,a) = \frac{\sum_{i=1}^{h} Q_i(s,a)}{h}$. Each head is trained separately with the standard cost function.

Empirical Evaluation

We evaluate RCMP in the *Pong* Atari game, where we use as demonstrator a previously-trained A3C agent. In our experiment, we compare: **RCMP**: Our proposal as described in the previous section; **No Advice**: Regular learning with no advice; **Random**: The agent receives advice randomly with no regard to its uncertainty; **Importance**: Importance advising as described in the Background section. Figure 2 shows the sum of undiscounted reward achieved by each algorithm with a maximum number of available advice set to $10,000$. Although stopping to ask for advice very early in the training, RCMP starts to show performance improvements over all other algorithms roughly around after 1.5 million of learning steps. All the advice-based algorithms outperform not receiving any advice, but our results show that estimating the agent uncertainty result in the best use of demonstrated data.

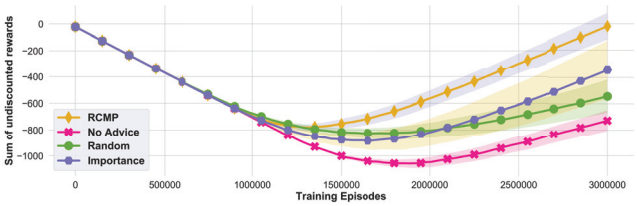

Figure 1: Illustration of a network with *heads*. Each head estimates a value for each action.

Figure 2: Sum of undiscounted rewards observed in 20 repetitions of the *Pong* experiment.

Conclusion and Further Work

We proposed *Requesting Confidence-Moderated Policy advice* (RCMP) to accelerate Reinforcement Learning (RL). Our method estimates the agent uncertainty and asks for advice from a demonstrator when the uncertainty is high, avoiding to take decisions where the value function is not expected to have converged yet. We show that RCMP performs better than learning without advice, with importance advising, and with randomly-timed advice in the Atari Pong domain. RCMP opens up several avenues for further work. The first one would be exploring alternative ways to estimate the agent uncertainty. RCMP could also be improved to make better used of the provided advice, such as devising a cost function moving towards the demonstrated behavior more quickly than just using it for exploration as we do here.

References

Amir, O., et al. 2016. Interactive Teaching Strategies for Agent Training. In *IJCAI*, 804–811.

Mnih, V., et al. 2015. Human-level Control through Deep Reinforcement Learning. *Nature* 518(7540):529–533.

Osband, I., et al. 2016. Deep Exploration via Bootstrapped DQN. In *NIPS*, 4026–4034.

Silva, F. L. D., and Costa, A. H. R. 2019. A Survey on Transfer Learning for Multiagent Reinforcement Learning Systems. *JAIR* 69:645–703.

Torrey, L., and Taylor, M. E. 2013. Teaching on a Budget: Agents Advising Agents in Reinforcement Learning. In *AAMAS*, 1053–1060.