Domain-Independent Optimistic Initialization for Reinforcement Learning

Marlos C. Machado, Sriram Srinivasan and Michael Bowling

Reinforcement Learning and Artificial Intelligence Laboratory,
Department of Computing Science, University of Alberta
{machado, ssriram, bowling}@ualberta.ca

Abstract

In Reinforcement Learning, it is common to use optimistic initialization of value functions to encourage exploration. However, such an approach generally depends on the domain, viz., the scale of the rewards must be known, and, when using function approximation, the feature representation must have a constant norm. We present a simple approach that performs optimistic initialization with less dependence on the domain.

Introduction

One of the challenges in Reinforcement Learning (RL) is the trade-off between exploration and exploitation. The agent must choose between taking an action known to give positive reward or to explore other possibilities hoping to receive a greater reward in the future. In this context, a common strategy in unknown environments is to assume that unseen states are more promising than those states already seen. One such approach is optimistic initialization of values (Sutton and Barto 1998, Section 2.7).

Several RL algorithms rely on estimates of expected values of states or expected values of actions in a given state (Sutton and Barto 1998). Optimistic initialization consists in initializing such estimates with higher values than are likely to be the true value. Also, when the state space is very large, it is common to represent each state as a feature vector with dimensions much smaller than the number of states. However, to be able to perform optimistic initialization, we depend on prior domain knowledge such as the expected scale of rewards and the structure of the feature vector. This paper circumvents such limitations presenting a different way to optimistically initialize value functions without additional domain knowledge or assumptions.

In the next section we formalize the problem setting as well as the RL framework. We then present our optimistic initialization approach. Also, we present some experimental analysis of our method using the Arcade Learning Environment (ALE) (Bellemare et al. 2013) as the testbed.

Problem Setting

Consider a Markov Decision Process, at time step t the agent is in a state $s_t \in \mathcal{S}$ and it needs to take an action $a_t \in \mathcal{A}$. Once the action is taken, the agent observes a new state s_{t+1} and a reward $r_{t+1} \sim R(s_t, a_t, s_{t+1})$ from a transition probability function $P(s_{t+1}|s_t, a_t) \equiv Pr(s_{t+1}|s_t, a_t)$. The agent's goal is to obtain a policy $\pi(a|s)$ that maximizes the expected discounted return $q_\pi(s_t, a_t) \equiv \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \Big| s_0, \pi\right]$, where $\gamma \in (0, 1]$ is the discount factor and $q_\pi(s, a)$ is the action-value function for policy π.

Sometimes it is not feasible to compute $q_\pi(s, a)$, we then approximate such values with linear function approximation: $q_\pi(s, a) \approx \theta^T \phi(s, a)$, where θ is a learned set of weights and $\phi(s, a)$ is the feature vector representing the current state. Note that $|\phi(s, a)| << |\mathcal{S}|$. However, as already mentioned, function approximation adds further difficulties for optimistic initialization, as one only indirectly specifies the value of state-action pairs through the choice of θ.

Optimistic Initialization

An approach to circumvent the requirement of knowing the reward scale is to normalize all rewards (r_t) by the first non-zero reward seen (r_{1st}), *i.e.*: $r_t/|r_{1st}|$. Then we can optimistically initialize $q_\pi(s, a)$ as 1, representing the expectation that a reward the size of the first reward will be achieved on the next timestep (or that a state with such value, after discounting, will be reached). Note that this is only a mild form of optimism. A more optimistic view might be that you can achieve reward on each step equal to that of the first observed reward, in which case we should aim to initialize $q_\pi(s, a)$ to $\frac{1}{1-\gamma}$. For sparse reward domains, which is common in the ALE, the mild form is often sufficient.

With function approximation, the approach above would mean initializing the weights θ to ensure $\theta^T \phi(s_t, a_t) = 1$, *e.g.*: $\theta_i = 1/|\phi(s_t, a_t)|$. However, this requires $|\phi(s_t, a_t)|$ to be constant among all states and actions. If the feature vector is binary-valued then one approach for guaranteeing ϕ has a constant norm is to stack $\phi(s_t, a_t)$ and $\neg\phi(s_t, a_t)$, where $\neg\phi(s_t, a_t)$ denotes the logical complement $\neg$ being applied to each coordinate of $\phi(s_t, a_t)$. While this achieves the goal, it has the cost of doubling the number of features. Besides, it removes sparsity in the feature vector, which can often be exploited for more efficient algorithms.

Our approach is to shift the value function so that a zero

function is in fact optimistic. We normalize by the first reward as described above. In addition, we shift the rewards downward by $\gamma - 1$, so $\tilde{r}_t = \frac{r_t}{|r_{1st}|} + (\gamma - 1)$. Thus, we have:

$$
\begin{aligned}
\tilde{q}_\pi(s_t, a_t) &= \mathbb{E}_\pi\left[\sum_{k=0}^\infty \gamma^k \tilde{r}_{t+k+1}\right] \\
&= \mathbb{E}_\pi\underbrace{\left[\sum_{k=0}^\infty \gamma^k \frac{r_{t+k+1}}{|r_{1st}|}\right]}_{\frac{q_\pi(s_t,a_t)}{|r_{1st}|}} + \underbrace{\sum_{k=0}^\infty \gamma^k(\gamma - 1)}_{-1}
\end{aligned}
$$

Notice that since $\tilde{q}_\pi(s_t, a_t) = \frac{q_\pi(s_t,a_t)}{|r_{1st}|} - 1$, initializing $\theta = 0$ is the same as initializing $q_\pi(s_t, a_t) = r_{1st}$. This shift alleviates us from knowing $|\phi(s,a)|$, since we do not have the requirement $\theta^T \phi(s, a) = 1$ anymore. Also, even though $\tilde{q}_\pi(s_t, a_t)$ is defined in terms of r_{1st}, we only need to know r_{1st} once a non-zero reward is observed.

In episodic tasks this shift encourages agents to terminate episodes as fast as possible to avoid negative rewards. To avoid this we provide a termination reward $r_{end} = \gamma^{T-k+1} - 1$, where k is the number of steps in the episode and T is the maximum number of steps. This is equivalent to receiving a reward of $\gamma - 1$ for additional $T - k + 1$ steps, and it forces the agent to look for a better alternative.

Experimental Analysis

We evaluated our approach in two different domains, with different reward scales and different number of active features. These domains were obtained from the ALE (Bellemare et al. 2013), a framework with dozens of Atari 2600 games where the agent has access, at each time step, to the game screen or the RAM data, besides an additional reward signal. Recently the ALE was used to extensively evaluate several different model-free linear learning algorithms, and optimistic initialization had been mentioned as an open problem in this environment (Defazio and Graepel 2014), due to some of the challenges already mentioned here.

We compare the learning curves of regular Sarsa(λ) (Rummery and Niranjan 1994) and Sarsa(λ) with its Q-values optimistically initialized. We used *Basic* features with the same Sarsa(λ) parameters reported by Bellemare et al.. The *Basic* features divide the screen in to 14×16 tiles and check, for each tile, if each of the 128 possible colours are active, totalling 28,672 features.

The results are presented in Figures 1-4. We report results using two different learning rates α, a low value ($\alpha = 0.01$) and a high value ($\alpha = 0.50$), each point corresponds to the average after 30 runs and they are reported in a sliding window of size 10, *i.e.* each point corresponds to the average score of the last 10 episodes.

The game FREEWAY consists in controlling a chicken that needs to cross a street, avoiding cars, to score a point (+1 reward). The episode lasts for 8192 steps and the agent's goal is to cross the street as many times as possible. This game poses an interesting exploration challenge for random exploration because it requires the agent to cross the street acting
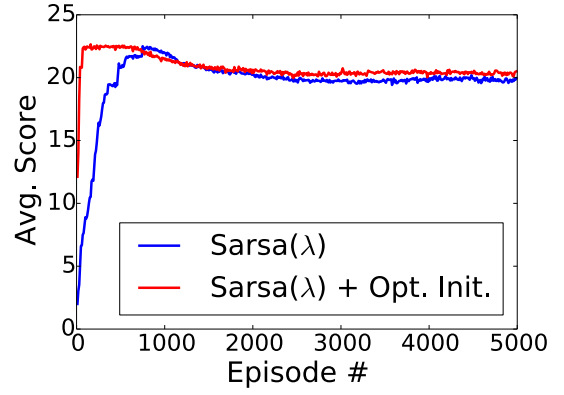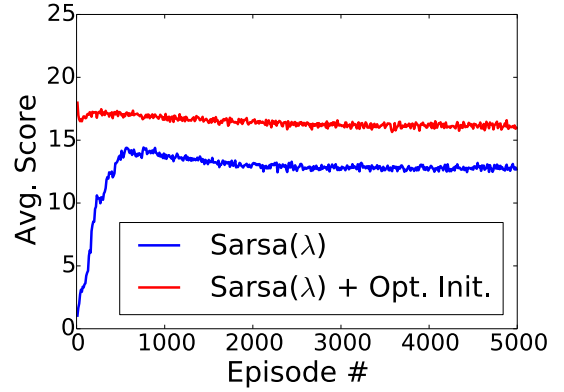

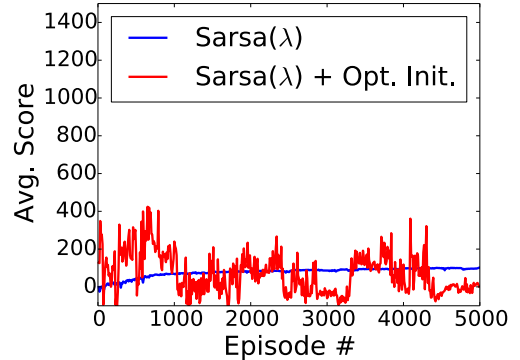Figure 1: FREEWAY; $\alpha = 0.01$

Figure 2: FREEWAY; $\alpha = 0.50$

Figure 3: PRIVATE EYE; $\alpha = 0.01$

randomly ($|\mathcal{A}| = 18$) for dozens of time steps. This means frequently selecting the action "go up" while avoiding cars. Looking at the results in Figure 1 and 2 we can see that, as expected, optimistic initialization does help since it favours exploration (*i.e.* states not seen, which are those closer to the other side of the street), speeding up the process of learning that a positive reward is available in the game. We see this improvement over Sarsa(λ) for both learning rates.
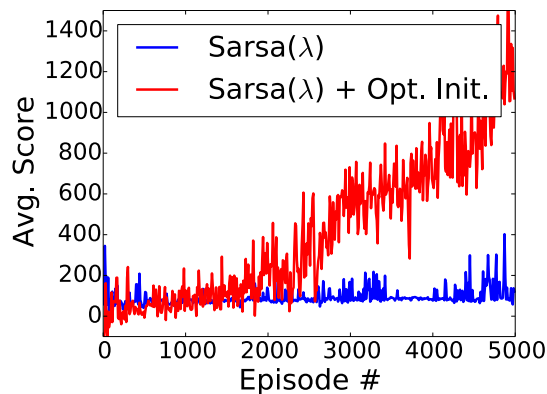
Figure 4: PRIVATE EYE; $\alpha = 0.50$

The game PRIVATE EYE is a very different domain. In this game the agent is supposed to move right for several screens (much more than when crossing the street in the game FREEWAY) and it should avoid enemies to avoid negative rewards. Along the path the agent can collect intermediate rewards ($+100$) but its ultimate goal is to get to the end and reach the goal, obtaining a much larger reward ($+5000$). We can see (Figures 3 and 4) that the optimistic initialization is much more reckless in the sense that it takes much more time to realize a specific state is not good (one of the main drawbacks of this approach), while Sarsa(λ) is more conservative. Interestingly, we observe that exploration may have a huge benefit in this game as a larger learning rate guides the agent to see rewards in a scale that was not seen by Sarsa(λ).

Thus, besides our formal analysis, we have shown that our approach behaves as one would expect optimistically initialized algorithms to behave. It increased agents' exploration with the trade off that sometimes the agent "exploited" a negative reward hoping to obtain a higher return. This was done without using any domain-specific information.

Conclusion

RL algorithms can be implemented without needing rigorous domain knowledge, but as far as we know, until this work, it was unfeasible to perform optimistic initialization in the same transparent way. Besides not requiring adaptations for specific domains, our approach does not hinder algorithms efficiency.

Acknowledgements

References

Bellemare, M. G.; Naddaf, Y.; Veness, J.; and Bowling, M. 2013. The Arcade Learning Environment: An Evaluation Platform for General Agents. *Journal of Artificial Intelligence Research* 47:253–279.

Defazio, A., and Graepel, T. 2014. A comparison of learning algorithms on the arcade learning environment. *CoRR* abs/1410.8620.

Rummery, G. A., and Niranjan, M. 1994. Online Q-learning using connectionist systems. CUED/F-INFENG/TR 166, Cambridge Univ. Engineering Dept.

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement Learning: An Introduction*. MIT Press.