FAKULTÄT FÜR WIRTSCHAFTSINFORMATIK
UND WIRTSCHAFTSMATHEMATIK

UNIVERSITÄT MANNHEIM

Prof. Dr. Leif Döring
André Ferdinand, Sara Klein

Reinforcement Learning

7. Excercise Sheet

1. Proof Theorem 4.2.9.

Close the gap in the proof of Theorem 4.2.9. from the lecture. Therefore go through the one dimensional proof of Theorem 4.2.8. and check that also in the d-dimensional case there exits a sequence $t_k \to \infty$ such that $\sup_{t \geq t_k} |x(t)| \leq D_k$ almost surely and $\lim_{k\to\infty} D_k = 0$.

2. SARSA

Rewrite a k-armed Bandit as a MDP in such a way that SARSA (Algorithm 25 with ϵ_n-greedy policy updates and $\alpha(s,a) = \frac{1}{N(s,a)+1}$) corresponds to the ϵ_n-greedy algorithm introduced in Chapter 1. Check that both algorithms are equivalent.

3. Convergence of Q-Learning

The assumptions and definitions of Theorem 4.3.4 (Convergence of Q-Learning) are given. Moreover let

$$F(Q)(s,a) \coloneqq \mathbb{E}_s^{\pi^a}\big[R(s,a) + \gamma \max_{a' \in \mathcal{A}_{S_1}} Q(S_1, a')\big]$$

and

$$\varepsilon_n(s,a) \coloneqq R(s,a) + \gamma \max_{a' \in \mathcal{A}_{s'}} Q_n(s',a') - F(Q_n)(s,a)$$

for all $(s,a) \in \mathcal{S} \times \mathcal{A}$ and $n \in \mathbb{N}$. Show that the sequence

$$Q_{n+1}(s,a) \coloneqq Q_n(s,a) + \alpha_n(s,a)\big(F(Q_n)(s,a) - Q_n(s,a) + \varepsilon_n(s,a)\big), n \in \mathbb{N}$$

almost surely converges to Q^π.

4. Maximization Bias Example

In this task we want to implement a Markov Decision process, which will show properties and difficulties of Q-Learning. The modeling of the constructed example comes from Example 6.7. The state space $\mathcal{S}$ is given by $\mathcal{S} = \{C, D, T\}$, where T is a terminal state. Furthermore, the possible actions in state C are given by $\mathcal{A}_C = \{\text{left}, \text{right}\}$ and $\mathcal{A}_D = \{0, 1, \ldots, n\}$, where $n \in \mathbb{N}$. The transition probabilities are given by $p(T; C, \text{right}) = 1, p(D; C, \text{right}) = 1$ and $p(D; T, a) = 1$ for all $a \in \mathcal{A}_D$. Moreover, the distribution of rewards is given by $R(S,A) \sim \mathcal{N}(-0.1, 1)$ if $S(\omega) = C, A(\omega) \in A_D, \omega \in \Omega$ and $R(S,A) \equiv 0$ otherwise.

5. One Step Approximate Dynamic Programming

In this task we want to implement the algorithms from chapter 4.3 (One-step approximate dynamic programming).

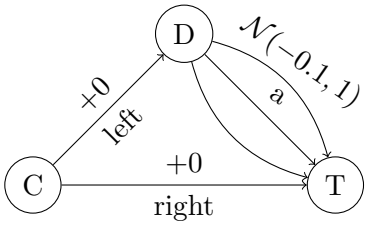

Abbildung 1: Visualization of the game from task 3. Action a is any action which can be played in state D. So $a \in A_D$.

(a) Implement Algorithm 22 (SARSA policy evaluation for Q^π). Replace the algorithm with the policy evaluation with Monte Carlo from the exercise sheet before and determine the optimal policy for one of the examples implemented so far.

(b) Implement Algorithm 23 (Q-Learning and SARSA) and determine the optimal policy for one of the examples implemented so far.

(c) Implement Algorithm 24 (SARSA - on-policy control for terminating MDPs) and determine the optimal policy for the Markov Decision Model from exercise 3.