FAKULTÄT FÜR WIRTSCHAFTSINFORMATIK
UND WIRTSCHAFTSMATHEMATIK

UNIVERSITÄT MANNHEIM

Prof. Dr. Leif Döring
Benedikt Wille

Reinforcement Learning

9. Exercise Sheet - Solutions

06.05.2025

1. Convergence of Q-Learning

The assumptions and definitions of Theorem 3.5.3 (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^*.

Solution:
We aim to apply Theorem 3.3.8.. Therefore we have to show that

a) $F : \mathbb{R}^{|\mathcal{S}||\mathcal{A}|} \to \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ *is a contraction with respect to the* $||\cdot||_\infty$*-norm, and*

b) $\varepsilon_n(s,a)$ *is* $\mathcal{F}_{n+1}$*-measurable,* $\mathbb{E}[\varepsilon_n(s,a)|\mathcal{F}_n] = 0$*, and there are some* A, B *such that* $\sup_{s,a} \mathbb{E}[\varepsilon_n^2(s,a)|\mathcal{F}_n] \leq A + B\|Q\|_\infty^2$.

We show a) by checking the definition of a contraction:

$$\begin{aligned}
&\|F(Q_1) - F(Q_2)\|_\infty \\
&= \max_{s,a} \Big\{ \big| \mathbb{E}_s^{\pi^a}\big[R(s,a) + \gamma \max_{a' \in \mathcal{A}_{S_1}} Q_1(S_1,a')\big] - \mathbb{E}_s^{\pi^a}\big[R(s,a) + \gamma \max_{a' \in \mathcal{A}_{S_1}} Q_2(S_1,a')\big] \big| \Big\} \\
&= \gamma \max_{s,a} \Big\{ \big| \mathbb{E}_s^{\pi^a}\big[\max_{a' \in \mathcal{A}_{S_1}} Q_1(S_1,a') - \max_{a' \in \mathcal{A}_{S_1}} Q_2(S_1,a')\big] \big| \Big\} \\
&\leq \gamma \max_{s,a} \Big\{ \big| \mathbb{E}_s^{\pi^a}\big[\max_{a' \in \mathcal{A}_{S_1}} (Q_1(S_1,a') - Q_2(S_1,a'))\big] \big| \Big\} \\
&\leq \gamma \max_{s,a} \Big\{ \mathbb{E}_s^{\pi^a}\big[\max_{s' \in \mathcal{S}, a' \in \mathcal{A}_{S_1}} \big|Q_1(s',a') - Q_2(s',a')\big|\big] \Big\} \\
&= \gamma \max_{s,a} \Big\{ \mathbb{E}_s^{\pi^a}\big[\|Q_1 - Q_2\|_\infty\big] \Big\} \\
&= \gamma \|Q_1 - Q_2\|_\infty.
\end{aligned}$$

We move on to claim b). The errors are $\mathcal{F}_{n+1}$-measurable by definition and for the expectation we see directly that

$$\begin{aligned}\mathbb{E}[\varepsilon_n(s,a)|\mathcal{F}_n] &= \mathbb{E}[R(s,a)+\gamma \max_{a'\in\mathcal{A}_{s'}} Q_n(s',a') - F(Q_n)(s,a)|\mathcal{F}_n] \\ &= \mathbb{E}\Big[R(s,a)+\gamma \max_{a'\in\mathcal{A}_{s'}} Q_n(s',a') - \mathbb{E}_s^{\pi^a}\left[R(s,a)+\gamma \max_{a'\in\mathcal{A}_{S_1}} Q_n(S_1,a')\right]\Big] \\ &= 0,\end{aligned}$$

because the state state s' in the algorithm is sampled from $p(\cdot\,;s,a)$. The last claim follows directly from the assumption on bounded rewards as in 3.5.1 and 3.5.2:

$$\begin{aligned}\mathbb{E}[\epsilon_s^2(n)|\mathcal{F}_n] &\le \mathbb{E}[(R(s,a)+\gamma \max_{a'\in\mathcal{A}_{s'}} Q_n(s',a'))^2] \\ &\le C^2 + 2\gamma C\|Q(n)\|_\infty + \gamma^2\|Q(n)\|_\infty^2 \le A + B\|Q(n)\|_\infty^2\end{aligned}$$

2. SARSA

Rewrite a k-armed Bandit as an MDP in such a way that SARSA (Algorithm 20 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.

Solution:
We define the state space to be $\mathcal{S} = \{1, T\}$ where 1 is the first state, the initial distribution is thus $\mu = \delta_1$, and T is the terminal state.
The action space is defined to be $\mathcal{A}_1 = \{1,\dots,k\}$ and $\mathcal{A}_T = \{N\}$ and can be interpreted as we play an arm between in $1,\dots,k$ in the state 1 and we do noting in the terminal state T.
Then we define the transition probabilities to be $p(\{T\};\{1\},a) = 1$ for all $a \in \mathcal{A}_1$.
The reward set $\mathcal{R}$ is given by the set of all possible rewards of all k arms united with a terminal reward $\{0\}$ whenever we are in the terminal state T and play action N. I.e. the rewards are defined to be independent of the states and whenever we play action $A_t = a \in \mathcal{A}_1$ the reward is distributed as the rewards of arm a of the bandit, $R_{t+1} = R(a) \sim P_a$ and whenever we play action $A_t = N$ the reward is defined to be $R_{t+1} = R(N) = 0$.
We choose $\gamma \in (0,1)$ arbitrarily, as γ will be irrelevant in the algorithm. Now recall the SARSA Algorithm with stepsizes chosen based on the visitation times and terminal states that are to be passed, as stated below in Algorithm 1.
For the initialisation of Q and N nothing changes. As we consider ϵ_n-greedy policies, consider a fixed sequence $(\epsilon_n)_{n\in\mathbb{N}_0}$ and initialise π with any ϵ_0-greedy policy, where we only have to consider state 1 as the action in state T is always N with probability one. As $Q \equiv 0$ we choose an arbitrary action (wlog action $a = 1$) with probability $1 - \frac{\epsilon_0(k-1)}{k}$ and all other actions $a' \in \mathcal{A}_1$ with probability $\frac{\epsilon_0}{k}$.
Next we enter the 'while not convergend'-loop and see that we initialise s always with 1, as we choose $\mu = \delta_1$. Then we choose $a \sim \pi(\cdot|1)$ after the ϵ_0-greedy policy defined above. As 1 is not

Result: Approximation $Q \approx Q^*$
Initialize $Q(s,a) = 0$ and $N(s,a) = 0$ for all $(s,a) \in S \times A$
Choose initial policy π.
while *not converged* **do**
  Initialize s
  Choose $a \sim \pi(\cdot\,;\, s)$
  while s *not terminal* **do**
    Take action a, sample reward $R(s,a)$ and next state s'.
    Choose $a' \sim \pi(\cdot \mid s')$.
    Determine step size α.
    $Q(s,a) = Q(s,a) + \alpha(R(s,a) + \gamma Q(s',a') - Q(s,a))$
    $N(s,a) = N(s,a) + 1$
    $s = s', a = a'$
    Choose policy π derived from updated Q-values.
  end
end

Algorithm 1: SARSA

a terminal state we take the action we sampled and recive a reward $R(1,a)$. Then we transit in the terminal state $s' = T$ almost surely and choose action $a' = N$ almost surely and update $Q(1,a)$, using $\alpha = \frac{1}{N(1,a)+1}$, and $N(1,a)$. As $s' = T$ is a terminal state we update the policy π as ϵ_1-greedy policy and continue again with initialising $s \sim \mu$ in the outer loop.
Observations:

- *We only fulfill the 'while s not terminal' condition once, i.e. this is not a real loop. Moreover we choose always $s' = T$ and $a' = N$.*
- *$Q(T,N)$ is never updated and stays 0 forever, i.e. together with the first observation we note that the term $\gamma Q(s',a')$ is zero forever.*
- *We only need to consider $Q(s,a)$ and $N(s,a)$ for $s = 1$, i.e. we can drop the dependence on s.*
- *We only need a policy in the state $s = 1$, i.e. we will only wrtie $\pi(\cdot)$ as a probability distribution of the possible arms.*
- *Sampling an action a after an ϵ-greedy policy is equivalent to sampling a uniform random variable $U \sim \mathcal{U}[0,1]$ and play the greedy action whenever $U > \epsilon$ or a uniformly choosen random action whenever $U \leq \epsilon$.*

All in all the algorithm simplifies to the Bandit-SARSA algorithm below.
Finally we observe that this algorithm equals the ϵ_n-greedy algorithm from Chapter 1 of the lecture, because for action a_n

$$\begin{aligned} Q^{new}(a_n) &= Q(a_n) + \frac{1}{N(a_n)+1}(R(a_n) - Q(a_n)) \\ &= \frac{N(a_n)}{N(a_n)+1} Q(a_n) + \frac{1}{N(a_n)+1} R(a_n). \end{aligned}$$

Result: Approximation $Q \approx Q^*$
Initialize $Q(a) = 0$ and $N(a) = 0$ for all $a \in \{1, \ldots, k\}$
Set $n = 0$
Set $\pi(\cdot) = \delta_1$ (choose any arm)
while *not converged* **do**
  Sample $U \sim \mathcal{U}[0, 1]$
  if $U \leq \epsilon_n$ **then**
    Choose a_n uniformly in $\{1, \ldots, k\}$
  else
    Choose $a_n \sim \pi(\cdot)$
  end
  Play arm a_n, observe reward $R(a_n)$.
  Determine stepsize $\alpha = \frac{1}{N(a_n)+1}$.
  $Q(a_n) = Q(a_n) + \alpha(R(a_n) - Q(a_n))$
  $N(a_n) = N(a_n) + 1$
  Set policy $\pi(\cdot)$ as ϵ_n greedy policy over the Q-values.
  $n = n + 1$
end

Algorithm 2: Bandit-SARSA

With the memory trick this is equivalent to

$$Q^{new}(a_n) = \frac{1}{N(a) + 1} \sum_{i=0}^{n} R(a_n) \mathbf{1}_{\{a_n = a\}},$$

which is the estimator of $\hat{Q}_a$ of arm a in the ϵ_n-greedy algorithm.