Prof. Dr. Leif Döring
Daniel Schmidt, Benedikt Wille **6. Exercise Sheet - Solutions**

Reinforcement Learning
13.04.2026

1. Proof of Lemma 2.4.6 for T-step MDPs

Prove Lemma 2.4.6 from the lecture by comparing with the discounted counterpart.
The following holds for the optimal time-state value function and the optimal time-state-action value function for any $s \in \mathcal{S}$:

(i) $V_t^*(s) = \max_{a\in\mathcal{A}_s} Q_t^*(s,a)$ for all $t \leq T$,

(ii) $Q_t^*(s,a) = r(s,a) + \sum_{s'\in\mathcal{S}} p(s'\,;\,s,a)V_{t+1}^*(s')$ for all $t < T$

In particular, V^* and Q^* satisfy the following Bellman optimality equations (backwards recursions):

$$V_t^*(s) = \max_{a\in\mathcal{A}_s}\Big\{r(s,a) + \sum_{s'\in\mathcal{S}} p(s';s,a)V_{t+1}^*(s')\Big\}, \quad s \in \mathcal{S},$$

and

$$Q_t^*(s,a) = r(s,a) + \sum_{s'\in\mathcal{S}} p(s';s,a)\max_{a'\in\mathcal{A}_{s'}} Q_{t+1}^*(s',a'), \quad s\in\mathcal{S}, a\in\mathcal{A}_s,$$

for all $t < T$.

Solution:

The Bellmann optimality equations are direct consequences of i) and ii) by simply plugging i) into ii) and vice-versa, so we will proceed by simply showing i) and ii) in this order. The proof is very similar to the proof of Theorem 2.1.23 but using the slightly more special definitions of Q_t and V_t as well as directly using only non-stationary policies. First we want to employ Proposition 2.4.4 to obtain

$$\sup_{\pi_t\in\Pi_t^t} V_t^\pi(s) = \max_{a\in\mathcal{A}_s} Q_t^\pi(s,a) \quad \forall t \leq T.$$

„$\geq$“ is trivial since the supremum over all kernels π_t at time t is of course bigger than the max over the deterministic kernels π_t choosing arm a at time t. The counterpart follows with Proposition 2.4.4 from the inequality

$$V_t^\pi(s) = \sum_{a\in\mathcal{A}_s} \pi_t(a\,;\,s)Q_t^\pi(s,a) \leq \max_{a\in\mathcal{A}_s} Q_t^\pi(s,a) \underbrace{\sum_{a\in\mathcal{A}_s} \pi_t(a\,;\,s)}_{=1} = \max_{a\in\mathcal{A}_s} Q_t^\pi(s,a).$$

Because of this and the fact that $Q_t^\pi(s,a)$ does not depend on π_t for any $\pi \in \Pi_t^T$ we obtain

$$\begin{aligned}\max_{a\in\mathcal{A}_s} Q_t^*(s,a) &= \max_{a\in\mathcal{A}_s} \sup_{\pi\in\Pi_t^T} Q_t^\pi(s,a)\\ &= \sup_{\pi\in\Pi_{t+1}^T} \max_{a\in\mathcal{A}_s} Q_t^\pi(s,a)\\ &= \sup_{\pi\in\Pi_{t+1}^T} \sup_{\pi_t\in\Pi_t^t} V_t^\pi(s)\\ &= \sup_{\pi\in\Pi_t^T} V_t^\pi(s)\\ &= V_t^*(s)\end{aligned}$$

for all $t < T$. Conversely, using the exact same trick as in the proof of Theorem 2.1.23 for the justification of the change of sum and supremum for all $t < T$ we obtain by Proposition 2.4.4:

$$\begin{aligned}Q_t^*(s,a) = \sup_{\pi\in\Pi_t^T} Q_t^\pi(s,a) &= \sup_{\pi\in\Pi_t^T} \Big(r(s,a) + \sum_{s'\in\mathcal{S}} p(s';s,a)V_{t+1}^\pi(s')\Big)\\ &= r(s,a) + \sum_{s'\in\mathcal{S}} p(s';s,a) \sup_{\pi\in\Pi_t^T} V_{t+1}^\pi(s')\\ &= r(s,a) + \sum_{s'\in\mathcal{S}} p(s';s,a)V_{t+1}^*(s').\end{aligned}$$

2. Example: T-step MDPs

Recall the Ice Vendor example from the lecture. Assume the maximal amount of ice cream is $m = 3$ and the damand distribution is given by $\mathbb{P}(D_t = d) = p_d$ with $p_0 = p_2 = \frac{1}{4}, p_1 = \frac{1}{2}$. Suppose the revenue function f, ordering cost function o and storage cost function h are given by

$$\begin{aligned}&f : \mathbb{N}_0 \to \mathbb{R},\ x \mapsto 9x,\\ &o : \mathbb{N}_0 \to \mathbb{R},\ x \mapsto 2x,\\ &h : \mathbb{N}_0 \to \mathbb{R},\ x \mapsto 2 + x.\end{aligned}$$

a) Set up the transition matrix $p(s_{t+1}; s_t, a_t)$ in a table, such that every $s_t + a_t$ maps to the probability to land in s_{t+1}, and the reward function $r(s_t, a_t, s_{t+1})$ for this example.

Solution:

The transition matrix is given as follows

$(s+a)\backslash s'$	0	1	2	3
0	1	0	0	0
1	$\frac{3}{4}$	$\frac{1}{4}$	0	0
2	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$	0
3	0	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$

The reward function $R(s_t, a_t, s_{t+1}) = f(s_t + a_t - s_{t+1}) - o(a_t) - h(a_t + s_t)$ is given by

$$R(s_t, a_t, s_{t+1}) = 9(s_t + a_t - s_{t+1}) - 2a_t - 2 - (s_t + a_t) = 8s_t + 6a_t - 9s_{t+1} - 2.$$

b) Calculate the expected reward $r(s,a)$ for every state action pair. Can you guess an optimal strategy for a one time step MDP?

Solution:

The expected reward is given by

$$r(s,a) = \sum_{r \in \mathcal{R}} r p(\mathcal{S} \times \{r\}\,;\, s,a) = \sum_{r \in \mathcal{R}} \sum_{s' \in \mathcal{S}} p(\{s'\} \times \{r\}\,;\, s,a) r$$
$$= \sum_{s' \in \mathcal{S}} p(s'; s,a) R(s,a,s'),$$

because the reward is deterministic for given s, a, s'. *The reward table is then*

$s \backslash a$	0	1	2	3
0	-2	$\frac{7}{4}$	1	-2
1	$\frac{15}{4}$	3	0	x
2	5	2	x	x
3	4	x	x	x

c) Suppose now you can play a 3-step MDP, hence you can order ice cream 3 times in $t = 0, 1, 2$. What is the optimal strategy for this finite time horizion MDP? Calculate the optimal state value, state-action value functions and the optimal policies using the greedy policy improvement algorithm from the lecture.

Hint: Use backward induction.

Solution:

We have as inition condition $V_3^* \equiv 0$ *and* $Q_2^* \equiv r$. *We follow from* Q_2^* *that the optimal policy is*

$$\pi_2^*(1;0) = 1, \quad \pi_2^*(0;1) = 1, \quad \pi_2^*(0;2) = 1, \quad \pi_2^*(0;3) = 1.$$

The value function $V_2^*(s) = \max_a Q_2^*(s,a)$, *are the red marked values in the reward tabel of b).*

It follows by

$$Q_1^*(s,a) = r(s,a) + \sum_{s' \in \mathcal{S}} p(s'; s,a) V_2^*(s')$$

that Q_1^* *is given by*

$s \backslash a$	0	1	2	3
0	$-\frac{1}{4}$	$\frac{61}{16}$	$\frac{67}{16}$	$\frac{9}{4}$
1	$\frac{93}{16}$	$\frac{99}{16}$	$\frac{17}{4}$	x
2	$\frac{131}{16}$	$\frac{25}{4}$	x	x
3	$\frac{33}{4}$	x	x	x

We follow from Q_1^* *that the optimal policy is*

$$\pi_1^*(2;0) = 1, \quad \pi_1^*(1;1) = 1, \quad \pi_1^*(0;2) = 1, \quad \pi_1^*(0;3) = 1.$$

The value function $V_1^*(s) = \max_a Q_1^*(s,a)$ *are the red numbers in the table. For the last timestep:*

$$Q_0^*(s,a) = r(s,a) + \sum_{s' \in \mathcal{S}} p(s'; s,a) V_1^*(s')$$

that Q_0^* is given by

$s\backslash a$	0	1	2	3
0	$\frac{35}{26}$	$\frac{413}{64}$	$\frac{231}{32}$	$\frac{203}{32}$
1	$\frac{605}{64}$	$\frac{295}{32}$	$\frac{331}{32}$	x
2	$\frac{359}{32}$	$\frac{331}{32}$	x	x
3	$\frac{395}{32}$	x	x	x

We follow from Q_1^* that the optimal policy is

$$\pi_0^*(2;0) = 1, \quad \pi_0^*(2;1) = 1, \quad \pi_0^*(0;2) = 1, \quad \pi_0^*(0;3) = 1.$$

Finally, we have that the red marked numbers in the last table are the optimal value function V_0^* of this MDP.

3. First visit Monte Carlo

Recall the first visit Monte Carlo Algorithm (14) from the lecture notes. Rewrite the estimate $V_n(s_t)$ to argue how we can apply the law of large numbers to show convergence (Hint: Use the strong Markov property).

Now consider the same algorithm without the if-condition in the for-loop. This algorithm is called every visit Monte Carlo algorithm (see Algorithm 1). Argue why we cannot apply the law of large numbers.

Data: Policy $\pi \in \Pi_S$, initial condition μ
Result: Approximation $\tilde{V} \approx V^\pi$
Initialize $V_0 \equiv 0$ and $N \equiv 1$
$n = 0$
while *not converged* **do**
  $n = n + 1$
  Sample $T \sim \mathrm{Geo}(1-\gamma)$.
  Sample s_0 from μ.
  Generate trajectory $(s_0, a_0, r_0, s_1, ...)$ until time horizon $2T$ using policy π.
  for $t = 0, 1, 2, ..., T$ **do**
    $v = \sum_{k=t}^{T+t} r_k$
    $V_n(s_t) = \frac{1}{N(s_t)+1} v + \frac{N(s_t)-1}{N(s_t)} V_{n-1}(s_t)$
    $N(s_t) = N(s_t) + 1$
  end
end
Set $\tilde{V} = V_n$.

Algorithm 1: Every visit Monte Carlo policy evaluation of V^π

Solution:
First, not that due to the strong Markov property we have that

$$\begin{aligned}&\mathbb{P}_\mu^\pi(S_t = s_t, A_t = a_t, R_t = r_t, \dots, S_{t+T}, A_{T+t}, R_{t+T} = r_{t+T} | S_0 = s_0, A_0 = a_0, \dots, S_t = s_t, A_t = a_t)\\ =&\mathbb{P}_\mu^\pi(R_0 = r_t, \dots, R_T = r_{t+T} | S_0 = s_t, A_0 = a_t)\end{aligned}$$

and thus in every algorithm step it holds $v \sim \sum_{k=0}^{T} R_k$ *given* $(S_0, A_0) = (s_t, a_t)$ *so that, denoting* v_n *as the realization of* v *in step* n *we see that these random variables are independent and identically distributed with expected value*

$$\mathbb{E}_{\mu}^{\pi}[v_n] = \mathbb{E}_{s_t}^{\pi}\left[\sum_{k=0}^{T} R_k\right] = V^{\pi}(s_t).$$

Applying the memory trick backwards we can thus write

$$V_n(s_t) = \frac{1}{N(s_t)+1} v_n + \frac{N(s_t)-1}{N(s_t)} V_{n-1}(s_t) = \frac{1}{N(s)} \sum_{k=1}^{N(s)} v_k,$$

which converges to V^{π} *due to the law of large numbers.*
For the every visit Monte Carlo algorithm the independence of the v_n *is not guaranteed, as in the event of multiple visits the random variables from the same rollout will be used multiple times. Thus, the law of large numbers can not be applied.*