Fakultät für Wirtschaftsinformatik
und Wirtschaftsmathematik

Universität Mannheim

Prof. Dr. Leif Döring

Daniel Schmidt, Benedikt Wille

Reinforcement Learning

02.03.2026

# 3. Exercise Sheet

*For this assignment, knowledge from lectures 1 to 8 is assumed.*

## 1. Markov Chains

Suppose that $(S_t)_{t\in\mathbb{N}}$ is a Markov Chain with values in some finite set $\mathcal{S}$ on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and denote by $P$ the transition matrix. Assume further that $\mathbb{P}(S_n = s') > 0$ for some $n \in \mathbb{N}$ and $s' \in \mathcal{S}$ and define the probability measure $\tilde{\mathbb{P}} = \mathbb{P}(\cdot \mid S_n = s')$. Prove that the shifted process $(\tilde{S}_t)_{t\in\mathbb{N}} = (S_{t+n})_{t\in\mathbb{N}}$ is again a Markov chain with transition matrix $P$.

## 2. Proof of Theorem 3.1.3

Complete the Proof of Theorem 3.1.3 in the lecture:

a) Show that defining $\mathbb{P}_T$ on the singletons

$$\mathbb{P}_T(\{(s_0, a_0, r_0, \ldots, s_T, a_T, r_T)\}) := \mu(\{s_0\}) \cdot \pi_0(\{a_0\}\,;\, s_0) \cdot \prod_{i=1}^{T} p(\{(s_i, r_{i-1})\}\,;\, s_{i-1}, a_{i-1})$$
$$\cdot\, \pi_i(\{a_i\}\,;\, s_0, a_0, \ldots, a_{i-1}, s_i) \cdot p(\mathcal{S} \times \{r_T\}; s_T, a_T).$$

yields a probability measure.

b) Check the claimed conditional probability identity

$$\mathbb{P}^\pi_\mu(S_{t+1} = s_{t+1}, R_t = r_t \,|\, S_t = s, A_t = a) = p(s_{t+1}, r_t\,;\, s, a).$$

where we assume that $\mathbb{P}^\pi_\mu(S_t = s, A_t = a) > 0$.

## 3. Probabilistic interpretation of MDPs

Use the formal definition of the stochastic process $(S, A, R)$ to check that

$$\begin{aligned}
p(s'; s, a) &= \mathbb{P}(S_{t+1} = s' \,|\, S_t = s, A_t = a),\\
p(r; s, a) &= \mathbb{P}(R_t = r | S_t = s, A_t = a),\\
r(s, a) &= \mathbb{E}[R_t \,|\, S_t = s, A_t = a],\\
r(s, a, s') &= \mathbb{E}[R_t \,|\, S_t = s, A_t = a, S_{t+1} = s']
\end{aligned}$$

holds if the events in the condition have positive probability, for any $t \in \mathbb{N}_0$.

4. **Proof of Proposition 3.1.13**

   Prove that if $\pi \in \Pi_S$ then $(S_t, A_t, R_t)_{t\in\mathbb{N}}$ satisfies the Markov reward process property

$$\begin{aligned}\mathbb{P}((S_{t+1}, A_{t+1}, R_{t+1}) = (s_{t+1}, a_{t+1}, r_{t+1}) \mid (S_t, A_t) = (s_t, a_t), \dots, (S_0, A_0) = (s_0, a_0))\\ = \mathbb{P}((S_{t+1}, A_{t+1}, R_{t+1}) = (s_{t+1}, a_{t+1}, r_{t+1}) \mid (S_t, A_t) = (s_t, a_t))\end{aligned}$$

   with time-homogeneous state/reward transition probabilities

$$p_{(s.a),(S',a',r')} = p(s', r; s, a)\pi(a'; s')$$

**The solution to the following exercise has to be turned in until 16.03.2026. Groups of up to three people are allowed.**

5. ***Programming task: Bandits**

   The aim of this exercise is to compare different bandit algorithms from the lecture. Your task is to run appropriate simulations and present meaningful plots that support your interpretations. Try to compare as many algorithms as possible, but at a minimum, include the following:

   - Explore-then-commit
   - $\epsilon$-greedy
   - UCB
   - Boltzmann Exploration
   - A policy gradient method

   a) What are the typical regret rates observed for each algorithm? Based on this, which algorithm performs best? Provide an intuitive explanation of how these rates arise and what constants should appear in front of them.

   b) Explain the effects of the exploration-exploitation tradeoff and committal behavior. Use appropriate graphs to illustrate these effects in your data. Discuss the differences between the algorithms in this context.

   c) In the bandit setting, what objectives besides regret minimization could be relevant (especially in the RL context)? Compare the algorithms based on these alternative objectives using appropriate metrics. Provide intuitive explanations of how these metrics arise and determine which algorithm performs best according to them. Consider the number of correctly chosen actions, estimates of arm means, and the probabilities of choosing the optimal arms over time as well as at the end of the time horizon.

   d) Use the optimal parameters from the lecture and think about how to estimate these parameters numerically for each algorithm. Compare your results in short to your findings in a)–c). For which algorithms does choosing the optimal parameters require „cheating“? How difficult is it to numerically determine the correct parameters? Based on your findings, which algorithm do you conclude is the best?

Remember to

- Indicate which Python version and packages you used.
- Label your graphs clearly.
- Specify the (hyper)parameter values chosen for your experiments and algorithms.

**The solution to the theoretical exercises will be discussed in the exercise class in B2 on March 09, 2026, at Mathelounge in B6 B301.**