Exercise 2

Kai Schneider

May 1, 2021

Task 1 Formulating Problems

as a Markov Decision Process $MDP = \{S, A, T, R, \gamma\}$

a.) chess

Discrete problem with a deterministic trasition function.

- **states:** all allowed configurations of the playing pieces (32, 16 for each player) on the field ($8x8 = 64$ segments). The number and type of the remaining pieces influences the possible configurations.
- **actions:** movements of the playing pieces. The number of available actions is determined by the current state (remaining pieces and reachable fields).
- **reward:** the only meaningful reward is determined by winning or loosing the game, since loosing playing pieces might be negative in the short run but benefit the long term strategy.

b.) pick & place robot

Continous problem (at least states and actions)

- **states:** all possible configurations of the joint angles the endeffector/toolhead ($s \in \mathbb{R}^n$)
- **actions:** all possible changes in joint angles and endeffector states ($a \in \mathbb{R}^n$)
- **reward:** multiple possibilities for a reward signal, e.g.:
  - positive reward for (succesfully) delivering a part
  - positive reward for moving with a part to the place location
    (and vice versa a neg. reward for moving without one)

c.) drone

State- and action-space are also continuous, although the drone itself surely operates discrete.

- **states:** 3D position and orientation (6 DOF) of the drone
  (for the controller also speed and acceleration)
- **actions:** changes/corrections in the rpm of the motors (assuming a multicopter-like drone)
- **reward:** reward could be a value relative to the deviation to the target value/position/orientation.

d.) own problem - commissioning

Picking and packing a predefined list of articles/objects from a larger range of things.

- **states:** the current state is always described by the already collected items (or in contrast all articles which still have to be collected).
- **actions:** Each a describes moving to another article and collecting it. A is the always the set of all a's for the remaining objects.
- **reward:** Like in the chess example, only the result after reaching the terminal state (collected all articles) is meaningful. Depending on the optimization goal a reward relative to the time (picking rate) or the covered distance (energie) might be a good choice.

Task 2 Value Functions

k-bandit:	$q(a) = \mathbb{E}[R_t \vert A_t = a]$
MDP:	$q(s,a) = \mathbb{E}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots \vert S_t = s, A_t = a]$

a.)

In contrast to the MDP, the bandits don't have multiple possible states. So each action A_t immediately returns a reward R_t. This reward doesn't depend on previous states/actions, so the potential future rewards aren't relevant here.

b.)

$$\begin{aligned}
\upsilon_\pi(s) = \mathbb{E}_\pi[G_t \vert S_t = s] &= \sum_a \Pi(a|s) \sum_{s'} \sum_r p(s', r|s, a)[r + \gamma \mathbb{E}[G_{t+1} \vert S_{t+1} = s']] \\
&\Leftrightarrow \sum_a \Pi(a|s) \sum_{s',r} p(s', r|s, a)[r + \gamma \sum_{a'} \Pi(a'|s') \mathbb{E}[G_{t+1} \vert S_{t+1} = s', A_{t+1} = a']] \\
&= \sum_a \Pi(a|s) \sum_{s',r} p(s', r|s, a)[r + \gamma \sum_{a'} \Pi(a'|s') q_\pi(s', a')] \\
&= \sum_a \Pi(a|s) q_\pi(s, a)
\end{aligned}$$

with slides 2.31 & 2.32 □

c.)

$$\begin{aligned}
\upsilon_\pi(s) &= \sum_a \Pi(a|s) \sum_{s',r} p(s', r|s, a)[r + \gamma \upsilon_\pi(s')] \\
&= \sum_a \Pi(a|s) \sum_{s'} \sum_r p(s', r|s, a)[r + \gamma \upsilon_\pi(s')] \\
&= \sum_a \Pi(a|s) \left[\sum_{s'} \sum_r p(s', r|s, a) r + \gamma \sum_{s'} \sum_r p(s', r|s, a) \upsilon_\pi(s') \right] \\
&= \sum_a \Pi(a|s) \left[\sum_{s'} p(s'|s, a) r(s, a, s') + \gamma \sum_{s'} p(s'|s, a) \upsilon_\pi(s') \right]
\end{aligned}$$

with slides 2.31 & 2.21 □