Q-learning

Exercise T14.1: Inductive value estimation and policy improvement (tutorial)

(a) How can value iteration be formulated as an online algorithm?

(b) What is *on-policy* and *off-policy* value estimation?

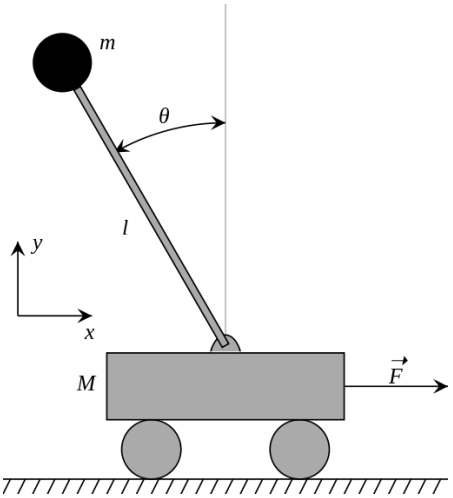
(c) How can the policy be improved?

(d) Derive the *Q-learning* algorithm.

(e) What is the *exploration-exploitation dilemma* and how can we address it?

Exercise H14.1: Inverted Pendulum (homework, 3 points)

In this exercise, you will implement a simplified version of the *inverted pendulum* task. Here a pole with a weight at the end is mounted on a cart (see image). The task is to balance the pole and keep it from falling down by moving the cart beneath it. However, for the sake of simplicity, we will ignore the cart here and apply force directly to the pole. The pole has a length of $l = 1\,m$ and a weight of $m = 2\,kg$. Earth has on its surface a gravity constant of $g = 9.81\,\frac{m}{s^2}$.



The system is characterized by two continuous state variables: the pole's *angle* θ and the *angular velocity* $\dot{\theta}$. We describe therefore the state as $\underline{\mathbf{x}}^{(t)} = [\theta_t, \dot{\theta}_t]^\top$. For a small time step $dt = 0.02\,s$, the dynamics are:

$$\begin{aligned} \dot{\theta}_{t+1} &= \dot{\theta}_t + \frac{g}{l}\sin(\theta_t)\,dt + \frac{F(\underline{\mathbf{a}}^{(t)})}{m}\,dt + \frac{\epsilon_t}{m}\,dt\,, \\ \theta_{t+1} &= \theta_t + \dot{\theta}_{t+1}\,dt\,, \qquad \epsilon_t \sim \mathcal{N}(0, \sigma^2)\,. \end{aligned}$$

Note that the pole is subject to strong pertubations ϵ_t, which are i.i.d. normal distributed with standard deviation $\sigma = 3\,N$. If left to its own devices, the pole will therefore fall.

To balance the pole, the agent can choose between 3 actions: to apply 4 Newtons of force in either direction, or to apply no force at all, i.e. $F(\underline{\mathbf{a}}_1) = -4\,N, F(\underline{\mathbf{a}}_2) = 0\,N, F(\underline{\mathbf{a}}_3) = +4\,N$. The experiment is performed in *episodes*: each episode starts at $\underline{\mathbf{x}}^{(0)} = [0, 0]^\top$, and continues for *either* 1000 time steps ($20\,s$) *or* until the absolute angle is larger than $\frac{\pi}{4}$, i.e. $|\theta_t| > \frac{\pi}{4}$. The latter is called a "failed episode", and the last transition of such an episode is punished by a reward of $r_t = -1$. All other transitions (with $|\theta_t| \leq \frac{\pi}{4}$) are not rewarded, i.e. $r_t = 0$.

(a) (2 points) Write a function that simulates one time step of the pole's dynamics. The function must predict the next state $\underline{\mathbf{x}}^{(t+1)}$, the reward earned in that period, and whether or not the episode has ended. Plot the angle θ for 10 uncontrolled episodes (i.e. with $F(\underline{\mathbf{a}}^{(t)}) = 0, \forall t$).

(b) (1 point) The continuous state space must be discretized in $D_1 \times D_2$ discrete states. Write a function that assigns a discrete state $s^{(t)} \in \{1, \ldots, D_1 \cdot D_2\}$ to each possible continuous state $\underline{\mathbf{x}}^{(t)}$. Assume that $-\frac{\pi}{4} \leq \theta_t \leq \frac{\pi}{4}$ and $-3 \leq \dot{\theta}_t \leq 3$. Draw randomly 100,000 continuous states from the normal distribution

$$\underline{\mathbf{x}}^{(t)} \quad \sim \quad \mathcal{N}\big(\underline{\mathbf{0}}, \underline{\mathbf{\Sigma}}^2\big)\,, \qquad \text{with standard deviations} \qquad \underline{\mathbf{\Sigma}} \quad = \quad \begin{bmatrix} \pi/8 & 0 \\ 0 & 3/2 \end{bmatrix},$$

and count how often the corresponding discrete states for $D_1 = D_2 = 50$ are drawn. Plot this count as a color-coded image with colorbar.

Exercise H14.2: Q-learning (homework, 7 points)

In this exercise you will implement and test Q-learning, i.e. the update rule:

$$\tilde{Q}^*_{t+1}(\underline{\mathbf{x}}^{(t)}, \underline{\mathbf{a}}^{(t)}) \quad = \quad \tilde{Q}^*_t(\underline{\mathbf{x}}^{(t)}, \underline{\mathbf{a}}^{(t)}) \quad + \quad \eta \underbrace{\Big(r_t + \gamma \max_{1 \leq k \leq 3} \tilde{Q}^*_t(\underline{\mathbf{x}}^{(t+1)}, \underline{\mathbf{a}}_k) - \tilde{Q}^*_t(\underline{\mathbf{x}}^{(t)}, \underline{\mathbf{a}}^{(t)}) \Big)}_{\text{TD-error } \Delta Q_t}$$

(a) (3 points) Implement Q-learning with an ε-greedy policy. Run your implementation for 2000 episodes with $D_1 = D_2 = 50, \varepsilon = 0, \eta = 0.5, \gamma = 0.9$ and an initialization of all Q-values with $Q_0 = 0$. Plot the number of steps your agent took in each episode. Additionally, plot both the final value function and the final greedy policy as image plots.

(b) (1 point) Repeat (a) with state discretizations of varying size, e.g. $D_1 = D_2 \in \{10, 50, 200\}$.

(c) (1 point) What is the smallest number of states that learns a very good policy?

(d) (1 point) Repeat (a), but change the ε-greedy policy to $\varepsilon = 0.1$.

(e) (1 point) Repeat (a), but change learning rate $\eta = 1$. Is the algorithm converging slower or faster to the close-to-optimal policy?

Total 10 points.