Ludwig-Maximilians-Universität München
Institut für Informatik
Prof. Dr. Matthias Schubert
Maximilian Bernhard, Niklas Strauß

Munich, 13.01.2025

Deep Learning and Artificial Intelligence

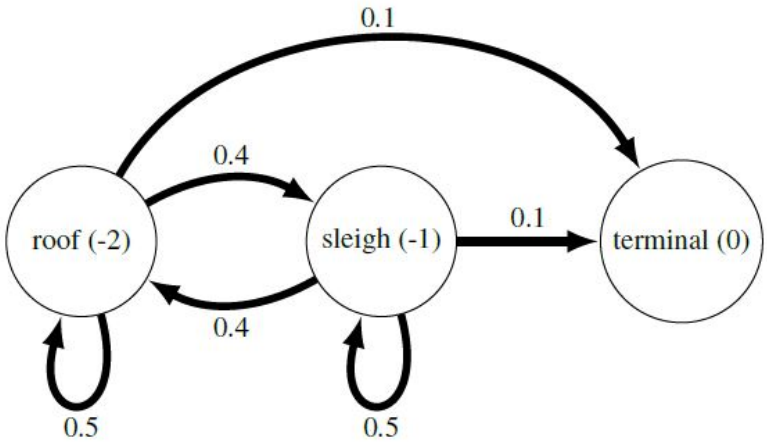
WS 2024/25

Exercise 10: Markov Decision Processes

Exercise 10-1 (Santa's) Markov Reward Process

On Christmas Eve, Santa has to deliver all presents to their recipients. To do so, he moves from house to house. Santa can try to throw a present into the chimney either directly from his sleigh or from the house's roof. After each throw, Santa decides that he has delivered enough presents and is done with the current house (terminal state) with probability 0.1. Otherwise, he switches his position from the sleigh to the roof or vice versa (probability 0.4) or stays put (probability 0.5).

Throwing the presents from the sleigh is a bit difficult, since Santa has to consider all winds and turbulences in the air. However, he is very experienced so that he will always hit the chimney successfully. The cost (negative reward) for throwing presents from the sleigh is thus -1. By contrast, the roofs are usually slippery and it may happen that Santa slips and falls to the ground so that he has to climb up again to deliver the present. Therefore, the cost for delivering presents from the roof is -2. The terminal state has a cost of 0. Below you can see Santa's (Markov) reward process representing Santa's work:



(a) Name the transition probability matrix P that is defined by the above Markov reward process.

$$\left[\text{Solution: } P = \begin{pmatrix} 0.5 & 0.4 & 0.1 \\ 0.4 & 0.5 & 0.1 \\ 0 & 0 & 0 \end{pmatrix}\right]$$

(b) Compute the expected discounted future rewards (utilities) for the states "roof", "sleigh" and "terminal" using the Bellman equation (with $\gamma = 1$):

$$U(s) = R(s) + \gamma \sum_{s'} P(s'|s)U(s'),\ \forall s \in S.$$

Detailed Solution

We can write this down as a linear equation system in matrix notation:

$$\begin{aligned} U &= R + \gamma P \cdot U \\ U - \gamma P \cdot U &= R \\ \underbrace{(I - \gamma P)}_{A}\underbrace{U}_{x} &= \underbrace{R}_{b} \\ \left[\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 0.5 & 0.4 & 0.1 \\ 0.4 & 0.5 & 0.1 \\ 0 & 0 & 0 \end{pmatrix}\right] \begin{pmatrix} u_s \\ u_r \\ u_t \end{pmatrix} &= \begin{pmatrix} -1 \\ -2 \\ 0 \end{pmatrix} \\ \begin{pmatrix} 0.5 & -0.4 & -0.1 \\ -0.4 & 0.5 & -0.1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u_s \\ u_r \\ u_t \end{pmatrix} &= \begin{pmatrix} -1 \\ -2 \\ 0 \end{pmatrix} \end{aligned}$$

Solving this, yields: $u_t = 0$, $u_r = -15.\overline{5}$ and $u_s = 30 - u_r = -14.\overline{4}$

Exercise 10-2 Policy Iteration

Santa wants to know whether he can do better. He can now decide at any time (not just after a throw) whether he wants to change his position (i.e. switch from the roof to the sleigh or vice versa, action c) or throw a present (action t). Due to the wind and the slippy roof, changing position only succeeds with probability 0.8. With probability 0.2, he just stays where he was. Moreover, each throw leads to Santa being done (reaching the terminal state) with probability 0.1 and staying put with probability 0.9.

Tasks:

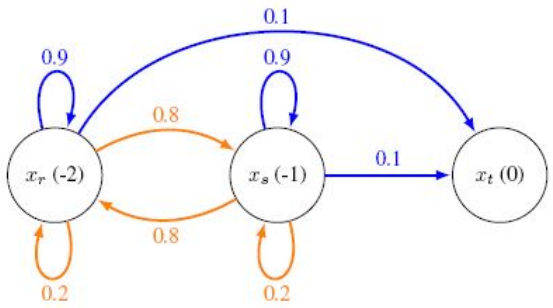
(a) Write down the transition probabilities for each action and state that are not 0 (you can also specify the transition matrix for each action). Moreover, draw the MDP described above, i.e. draw a node for each state and an edge for each possible transition. Use different colors for the two actions and annotate the edges with their transition probabilities and the nodes with their rewards.

Detailed Solution

Denoting the states as x_s, x_r and x_t, we have

$$\begin{aligned} \mathrm{P}(\mathrm{x}_r|x_s, c) &= 0.8 & P(x_s|x_r, c) &= 0.8 \\ \mathrm{P}(\mathrm{x}_s|x_s, c) &= 0.2 & P(x_r|x_r, c) &= 0.2 \\ \mathrm{P}(\mathrm{x}_t|x_s, t) &= 0.1 & P(x_t|x_r, t) &= 0.1 \\ \mathrm{P}(\mathrm{x}_s|x_s, t) &= 0.9 & P(x_r|x_r, t) &= 0.9 \end{aligned}$$

The MDP looks as follows:



(b) What can be said intuitively about the optimal policy in states x_s and x_r?

Detailed Solution

Intuitively, Santa wants to get to the terminal state x_t as soon as possible since he will pay a negative reward for each time step he spends in state x_s or x_r. However, action t leads to the terminal state with low probability. Santa should thus try to minimize the cost he pays (his effort) while trying to reach the terminal state. This suggests that if he is on the sleigh, he should stay put and keep throwing presents from there (i.e. perform action t). By contrast, if he is on the roof, it might be better if he tried to get onto his sleigh first (action c), which is the better place from which to throw presents since it has lower cost.

(c) Apply policy iteration to determine the optimal policy and the state-values of x_s and x_r. Assume the initial policy π_0 has action t in both states. For the policy-evaluation step, use the simplified Bellman equation for a fixed policy π_t:

$$U^{\pi_t}(x) = U^t(x) = R(x) + \gamma \sum_{x'} P(x' |\, x, \pi_t(x))\, U^t(x').$$

Why does Santa have such a big belly?

Detailed Solution

We start with policy evaluation:

$$\begin{aligned} u_s &= -1 + 0.1u_t + 0.9u_s \\ u_r &= -2 + 0.1u_t + 0.9u_r \\ u_t &= 0 \end{aligned}$$

If we plug in $u_t = 0$ into the first two equations, we get: $u_s = -10$ and $u_r = -20$.

In the policy improvement step, we calculate:

$$\begin{aligned} \pi_{t+1}(x) &= \underset{a \in A(x)}{\operatorname{argmax}} [R(x) + \gamma \sum_{x'} P(x' |\, x, a)\, U(x')] \\ &= \underset{a \in A(x)}{\operatorname{argmax}} \sum_{x'} P(x' |\, x, a)\, U(x') \end{aligned}$$

In the last step we used the fact that the R and γ are not dependent on the action and thus do not affect the maximization problem. Hence, they can be neglegted. Denoting $\sum_{x'} P(x' |\, x, a)\, U(x')$ as $T(x_i, a)$, we calculate:

$$\begin{aligned} T(x_s, c) &= P(x_r \,|x_s, c) \cdot u_r + P(x_s \,|x_s, c) \cdot u_s = 0.8 \cdot (-20) + 0.2 \cdot (-10) = -18 \\ T(x_s, t) &= P(x_t \,|x_s, t) \cdot u_t + P(x_s \,|x_s, t) \cdot u_s = 0.1 \cdot 0 + 0.9 \cdot (-10) = -9 \\ T(x_r, c) &= 0.8 \cdot (-10) + 0.2 \cdot (-20) = -12 \\ T(x_r, t) &= 0.1 \cdot 0 + 0.9 \cdot (-20) = -18 \end{aligned}$$

Since $T(x_s, c)T(x_s, t)$, action t is preferred in x_s. However, in state x_r, action c is preferred. Thus, we have $\pi_1(x_s) = t$ and $\pi_1(x_r) = c$. Since the policy for x_r changed, we set $unchanged?$ to false and proceed.

Policy evaluation:

$$\begin{aligned} u_s &= -1 + 0.1u_t + 0.9u_s \\ u_r &= -2 + 0.8u_s + 0.2u_r \\ u_t &= 0 \end{aligned}$$

Solving this linear equation system yields $u_s = -10$ and $u_r = -12.5$.
Note: To calculate the new utilities U_{new}, policy evaluation takes policy π, old utilities U_{old}, and MDP as inputs. Because the policy π depends on the old utilities U_{old}, so U_{old} is incorporated into the policy evaluation.

Policy improvement:

$$\begin{aligned} T(x_s, c) &= 0.8 \cdot (-12.5) + 0.2 \cdot (-10) && = -12 \\ T(x_s, t) &= 0.1 \cdot 0 + 0.9 \cdot (-10) && = -9 \\ T(x_r, c) &= 0.8 \cdot (-10) + 0.2 \cdot (-12.5) && = -10.5 \\ T(x_r, t) &= 0.1 \cdot 0 + 0.9 \cdot (-12.5) && = -11.25 \end{aligned}$$

Therefore, $\pi_2(x_s) = t$ and $\pi_2(x_r) = c$. The boolean $unchanged?$ remains true and we terminate.
$\Rightarrow$ Santa has such a big belly since the best strategy for him is to throw all presents directly from the sleigh instead of climbing onto the roof.

(d) What happens if the initial policy has action c in both states? Does discounting help?

Detailed Solution

An initial policy with action *c* for both states leads to an unsolvable problem. In the first policy evaluation step we get the linear equation system:

$$\begin{aligned} u_s &= -1 + 0.2u_s + 0.8u_r \\ u_r &= -2 + 0.8u_s + 0.2u_r \\ u_t &= 0 \end{aligned}$$

in which the first two equations are inconsistent. If we tried to solve them by value iteration, we would find the values tending to -∞. Discounting leads to well-defined solutions by bounding the expected discounted reward an agent can incur at either state. The above equations can be then reformulated as:

$$\begin{aligned} u_s &= -1 + 0.2 \cdot \gamma \cdot u_s + 0.8 \cdot \gamma \cdot u_r \\ u_r &= -2 + 0.8 \cdot \gamma \cdot u_s + 0.2 \cdot \gamma \cdot u_r \\ u_t &= 0 \end{aligned}$$

which makes the problem solvable.

Exercise 10-3 MDPs with Python

You will find a Jupyter notebook on Moodle that contains a programming exercise on MDPs. Please follow the instructions in the notebook file.