FAKULTÄT FÜR WIRTSCHAFTSINFORMATIK
UND WIRTSCHAFTSMATHEMATIK

UNIVERSITÄT MANNHEIM

Prof. Dr. Leif Döring
André Ferdinand, Sara Klein

Reinforcement Learning

4. Excercise Sheet

1. Proof of Theorem 3.1.3

Complete the Proof of Theorem 3.1.3. from the lecture:

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

$$\begin{aligned}&\mathbb{P}_T(\{(s_0, r_0, a_0, \dots, s_T, r_T, a_T)\})\\ &:= \mu(\{s_0\}) \cdot \delta_0(r_0) \cdot \pi_0(\{a_0\}\,;\, s_0) \cdot \prod_{i=1}^{T} p(\{(s_i, r_i)\}\,;\, s_{i-1}, a_{i-1}) \cdot \pi_i(\{a_i\}\,;\, s_0, a_0, \dots, a_{i-1}, s_i).\end{aligned}$$

yields a probability measure.

b) Check the claimed conditional probability identity

$$\mathbb{P}_\mu^\pi(S_{t+1} \in B, R_{t+1} \in D \,|\, S_t = s, A_t = a) = p(B \times D\,;\, s, a).$$

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

2. Probabilistic interpretation of MDPs

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

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

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

3. Bellman expectation equation for Q

Derive the Bellman expectation equation for the Q-function of a policy π:

$$Q^\pi(s, a) = r(s, a) + \gamma \sum_{s' \in \mathcal{S}} \sum_{a' \in \mathcal{A}} p(s'\,;\, s, a)\pi(a'\,;\, s) Q^\pi(s', a')$$

4. Programming task, Gumbel trick extended

On the previous exercise sheet, we looked at the Boltzmann Exploration and the relationship between the Softmax distribution and the Gumbel distribution. In this exercise you should try other random variables instead of the Gumbel distribution for the algorithmus and think about what a good or bad choice of random variables can be.
The following approach might be useful.

(a) Implement the algorithm 1 more generally by allowing them to pass any random variable instead of the Gumbel distribution. This code might be useful.

(b) Look in scipy.stats to see which random variables are implemented.

(c) Try the following random variables: Cauchy, Beta, Betaprime, Chi

Input : Initialization $\hat{Q}_a(0), a \in \mathcal{A}$, number of total timesteps n, number of arms $k \in \mathbb{N}$ and parameter $C \in \mathbb{R}$

Output: Trajectory of Rewards $(X_t)_{t\in\{1,\dots,n\}}$ and actions $(A_t)_{t\in\{1,\dots,n\}}$

begin

  for $t \leftarrow 1$ **to** n **do**

    Simulate $z_a, a \in \mathcal{A}$ independently identically standard Gumbel;

    Set $a_t = \arg\max_{a\in\mathcal{A}} \{\hat{Q}_a(t) + \sqrt{\frac{C^2}{N_a}} z_a\}$;

    Obtain reward x_t by playing arm a_t;

    Set $N_{a_t} = N_{a_t} + 1$;

    Update the estimated action value functions $\big(\hat{Q}_a(t)\big)$;

  end

  return $(X_t)_{t\in\{1,\dots,n\}}, (A_t)_{t\in\{1,\dots,n\}}$;

end

Algorithm 1: Boltzmann exploration algorithm modified