Prof. Frank Hutter, Prof. Joschka Bödecker

Dr. Marius Lindauer
Gabriel Kalweit

Reinforcement Learning
Exercise 3
Submit until **Thursday, November 30 at 2:00pm**

Before we learn how to use the methods from this week for control – we actually implement this next week –, we first have to understand the basic concepts. So, this week is a mix of theory and practice. Please push your solutions to subdirectory `exercise-03` in your assigned git-repository. We are going to submit a `feedback.txt` in that directory.

Preliminaries

This exercise is based on Lecture 4[1] from David Silver's RL course[2]. Watch before the upcoming meeting on Friday, November 24.

1 Monte Carlo and TD(λ) (10p)

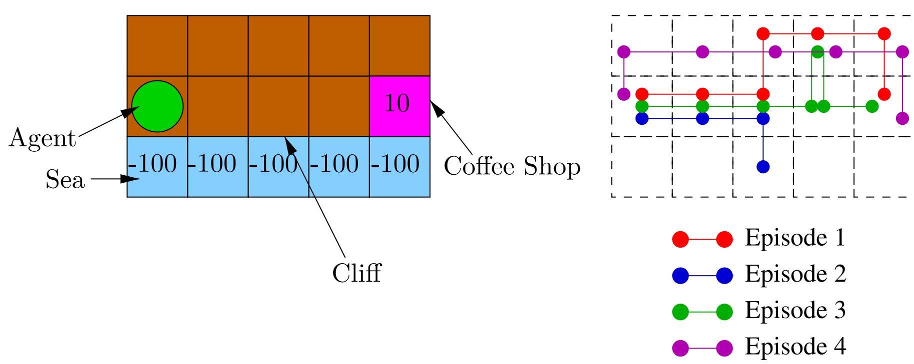

Figure 1: Cliff MDP

Consider the MDP in Figure 1, where all actions (an action moves the agent in a desired direction: N,S,E,W) succeed with a probability of 0.8. With a probability of 0.2 the agent moves randomly in another direction. All transitions result in a reward of -1, except when the coffee shop is reached (terminal state $s_{2,5}$: reward of 10) or if the agent falls of the cliff (terminal states $s_{3,1} \dots s_{3,5}$: reward of -100). The agent always starts in the start state $s_{2,1}$ as indicated in Figure 1.

[1] `https://youtu.be/PnHCvfgC_ZA`
[2] `http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching.html`

(a) Using Monte-Carlo policy evaluation, calculate $V_3(i)$ for all states i based on the illustrated episodes 1 to 3 (right part of Figure 1). Use the first-visit-method, i.e. every state is updated only once – on the first-visit – per episode, even if the state is visited again during the episode. In this task, we estimate the value by a running mean with $\alpha_t = \frac{1}{t}$ for episode t and $V_0(i) = 0$ for all i. We do not use discount, i.e. $\gamma = 1$.

(b) Consider now Episode 4 (magenta). Specify for all states visited during this episode the Temporal Difference error based on the value-function $V_3(\cdot)$ calculated in (a).

(c) Using the TD(λ)-algorithm, determine for $\lambda = 0$, $\lambda = 0.5$ and $\lambda = 1.0$ the expected value $v_\pi(s_{2,1})$ based on the first three episodes.

2 First-visit MC Evaluation (10p)

Implement the First-visit MC Evaluation algorithm introduced in the first part of Lecture 4,

```
mc_evaluation(policy, env, num_episodes, discount_factor=1.0)
```

in `YOUR_REPO/exercise-03/scripts/mc_evaluation.py`, where

- `policy` is a function that maps an observation to action probabilities and
- `env` is an OpenAI gym environment.

It returns a dictionary that maps from state to value.

This task is based on the Blackjack example from the lecture[3] and an implementation can be found at `lib.envs.blackjack`. The state is a tuple – containing the players current sum, the dealer's one showing card (1-10 where 1 is ace) and whether or not the player holds a usable ace (0 or 1) – and the value is a float. You find the tests at `YOUR_REPO/exercise-03/tests/exercise-03_test.py`. Run them with

```
python exercise-03_test.py -v
```

or with

```
python -m unittest exercise-03_test.py -v.
```

In addition, in `YOUR_REPO/exercise-03/scripts` you also find a visualization script of the predicted value-functions for which you need `matplotlib`[4]. You can run it with

```
python mc_evaluation_visualization.py.
```

3 Bonus: Experiences (1p)

Submit an `experiences.txt`, where you provide a brief summary of your experience with this exercise, the corresponding lecture and the last meeting. As a minimum, say how much time you invested and if you had major problems – and if yes, where.

Please push your solutions to subdirectory `exercise-03` in your assigned git-repository by **Thursday, November 30 at 2:00pm**. **Solutions after that or via email will not be accepted.**

[3] `http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching_files/MC-TD.pdf#page=8`
[4] `https://matplotlib.org/users/installing.html`