Prof. Frank Hutter, Prof. Joschka Bödecker

Dr. Marius Lindauer
Gabriel Kalweit

Reinforcement Learning
Exercise 2
Submit until **Thursday, November 23 at 2:00pm**

This week, we provide code snippets that are to be filled by you. Please follow the coding instructions in each task. You will also find tests you can check against. When the tests run through, you know the results are correct. **Please comment on your code and use meaningful variable names.** Blocks of code that are non-trivial need to be commented in order to be seen as correct. Explain what you are doing. This will also help you to get a deeper understanding of the concepts. Please push your solutions to subdirectory `exercise-02` in your assigned git-repository. We are going to submit a `feedback.txt` in that directory.

Preliminaries

This exercise is based on Lecture 3[1] from David Silver's RL course[2]. Watch before the upcoming meeting on Friday, November 17.

1 Dynamic Programming (20p)

The tests for the following tasks are based on the Gridworld environment from Sutton's Reinforcement Learning book chapter 4[3]. The agent moves on an $m \times n$ grid and the goal is to reach one of the terminal states at the top left or the bottom right corner. A visualization can be seen in Figure 1.

$$\begin{bmatrix} T & \cdot & \cdot & \cdot \\ \cdot & A & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & T \end{bmatrix}$$

Figure 1: An example of a 4×4 grid. Terminal states T and agent A.

The agent can go *up*, *down*, *left* and *right*. Actions leading off the edge do not change the state. The agent receives a reward of -1 in each step until it reaches a terminal state. An implementation of this environment is given in `lib.envs.gridworld`.

[1] `https://youtu.be/Nd1-UUMVfz4`
[2] `http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching.html`
[3] `http://incompleteideas.net/sutton/book/bookdraft2017nov5.pdf#page=79`

You find the tests at `YOUR_REPO/exercise-02/tests/exercise-02_test.py`. Run them with

```
python exercise-02_test.py -v
```

or with

```
python -m unittest exercise-02_test.py -v.
```

1.1 Policy Iteration (12p)

(a) Implement the Policy Evaluation function,

```
policy_eval(policy, env, discount_factor=1.0, theta=0.00001)
```

in `YOUR_REPO/exercise-02/scripts/policy_iteration.py`, where

- `policy` is a $[S, A]$ ($\#S$ states and $\#A$ actions) shaped matrix representing the policy,
- `env` is a discrete OpenAI environment and `env.P[s][a]` is a transition tuple (transition probability, next_state, reward, done) for state s and action a, and
- `theta` is the stopping threshold. We stop the evaluation once our value-function change (difference between two iterations) is less than `theta` for all states.

It returns a vector of length S representing the value-function. (6p)

(b) Implement the Policy Improvement function,

```
policy_improvement(env, policy_eval_fn=policy_eval, discount_factor=1.0)
```

in `YOUR_REPO/exercise-02/scripts/policy_iteration.py`. It returns a tuple `(policy, V)` where `policy` is the optimal policy – a matrix of shape $[S, A]$ where each state s contains a valid probability distribution over actions – and V is the value-function for the optimal policy. (6p)

1.2 Value Iteration (6p)

(a) Implement the Value Iteration function,

```
value_iteration(env, theta=0.0001, discount_factor=1.0)
```

in `YOUR_REPO/exercise-02/scripts/value_iteration.py`. It again returns a tuple `(policy, V)` of the optimal policy and the optimal value-function. (6p)

(b) What is the difference between Value Iteration and Policy Iteration? Compare the two methods. Submit a PDF or text file. (2p)

2 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-02` in your assigned git-repository by **Thursday, November 23 at 2:00pm**. **Solutions after that or via email will not be accepted.**