Exercise 12

Deadline: 21.07.2017, 2:15 pm
Regulations: You should hand in the exercises in groups of two or three persons. Please send a *compressed* (!) directory or file containing your solutions including all graphics, descriptions and source code to *thorsten.beier@iwr.uni-heidelberg.de*. The subject line of this email should start with [MLCV17][EX12] followed by the full names of all group members. Please cross-reference your code files in your writeup, such that it is clear which file has to be run for each exercise.

1 Reinforcement Learning

This exercise is based on the excellent "Practical RL" course from the Yandex School of Data Analysis [1].

To solve this exercise follow the instructions in the jupyter notebook. [2] A copy of the notebook is also attached to this pdf.

[1] `https://github.com/yandexdataschool/Practical_RL/`
[2] `https://github.com/DerThorsten/mlcv/blob/master/e12/crossentropy_method.ipynb`

crossentropy_method

July 14, 2017

1 References

This notebook is heavily based on the excellent "Practical RL" course from the Yandex School of Data Analysis https://github.com/yandexdataschool/Practical_RL/

2 Crossentropy method

This notebook will teach you to solve reinforcement learning with crossentropy method.

```
In [ ]: #XVFB will be launched if you run on a server
        import os
        if type(os.environ.get("DISPLAY")) is not str or len(os.environ.get("DISPLAY"))==0:
            !bash ../xvfb start
            %env DISPLAY=:1
```

```
In [ ]: import gym
        import numpy as np, pandas as pd

        env = gym.make("Taxi-v2")
        env.reset()
        env.render()
```

```
In [ ]: n_states = env.observation_space.n
        n_actions = env.action_space.n

        print("n_states=%i, n_actions=%i"%(n_states,n_actions))
```

3 Create stochastic policy

This time our policy should be a probability distribution.
`policy[s,a] = P(take action a | in state s)`
Since we still use integer state and action representations, you can use a 2-dimensional array to represent the policy.
Please initialize policy **uniformly**, that is, probabililities of all actions should be equal.

```
In [ ]: policy = <your code here! Create an array to store action probabilities>
```