VU Machine Learning

Winter Term 2022

Exercise 3.3

Nysret Musliu

This is one of possible topics for exercise 3. See other possible topics from my colleague in tuwel. You have to select only one topic for exercise 3

Exercise 3.3.

Your can select one of topics below (see Reinforcement Learning: An Introduction, R. Sutton and A. Barto http://incompleteideas.net/book/RLbook2020.pdf)

3.3.1 k-armed Bandit Problem: Modified Exercise 2.5, page 33 (see next page)

This topic has been considered in the class

OR

3.3.2 Exercise 5.12: Racetrack (programming), page 111.

The Monte Carlo method has not been considered in the class. Therefore, the selection of this assignment would require additional self-study

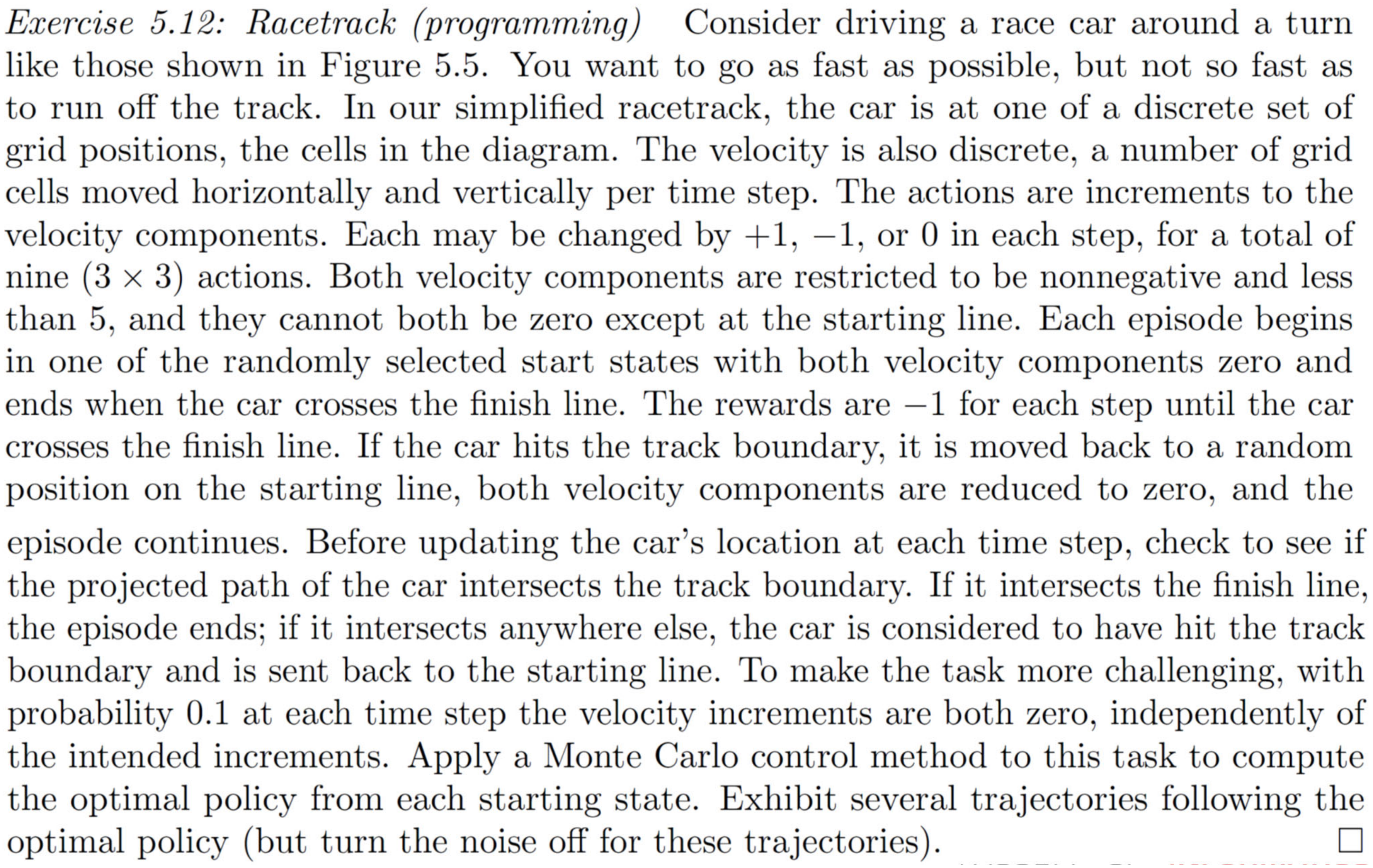
3.3.1 k-armed Bandit Problem

RL Book, page 33:

(Extended) Exercise 2.5 (programming) :

Design and conduct an experiment to demonstrate the difficulties that sample-average methods have for nonstationary problems. Use a modified version of the 10-armed testbed in which all the $q_*(a)$ start out equal and then take independent random walks (say by adding a normally distributed increment with mean zero and standard deviation 0.01 to all the $q_*(a)$ on each step). Prepare plots like Figure 2.2 (page 29) for an action-value method using sample averages, incrementally computed, and another action-value method using a constant step-size parameter, α = 0.1. Use ε= 0.1 and longer runs, say of 10,000 steps. Experiments also with Upper-Confidence-Bound Action Selection (section 2.7) and show the results of your experiments.

3.3.2. Racetrack (RL Book, page 111)



Exercise 5.12: Racetrack (programming) Consider driving a race car around a turn like those shown in Figure 5.5. You want to go as fast as possible, but not so fast as to run off the track. In our simplified racetrack, the car is at one of a discrete set of grid positions, the cells in the diagram. The velocity is also discrete, a number of grid cells moved horizontally and vertically per time step. The actions are increments to the velocity components. Each may be changed by +1, −1, or 0 in each step, for a total of nine (3 × 3) actions. Both velocity components are restricted to be nonnegative and less than 5, and they cannot both be zero except at the starting line. Each episode begins in one of the randomly selected start states with both velocity components zero and ends when the car crosses the finish line. The rewards are −1 for each step until the car crosses the finish line. If the car hits the track boundary, it is moved back to a random position on the starting line, both velocity components are reduced to zero, and the episode continues. Before updating the car's location at each time step, check to see if the projected path of the car intersects the track boundary. If it intersects the finish line, the episode ends; if it intersects anywhere else, the car is considered to have hit the track boundary and is sent back to the starting line. To make the task more challenging, with probability 0.1 at each time step the velocity increments are both zero, independently of the intended increments. Apply a Monte Carlo control method to this task to compute the optimal policy from each starting state. Exhibit several trajectories following the optimal policy (but turn the noise off for these trajectories). □

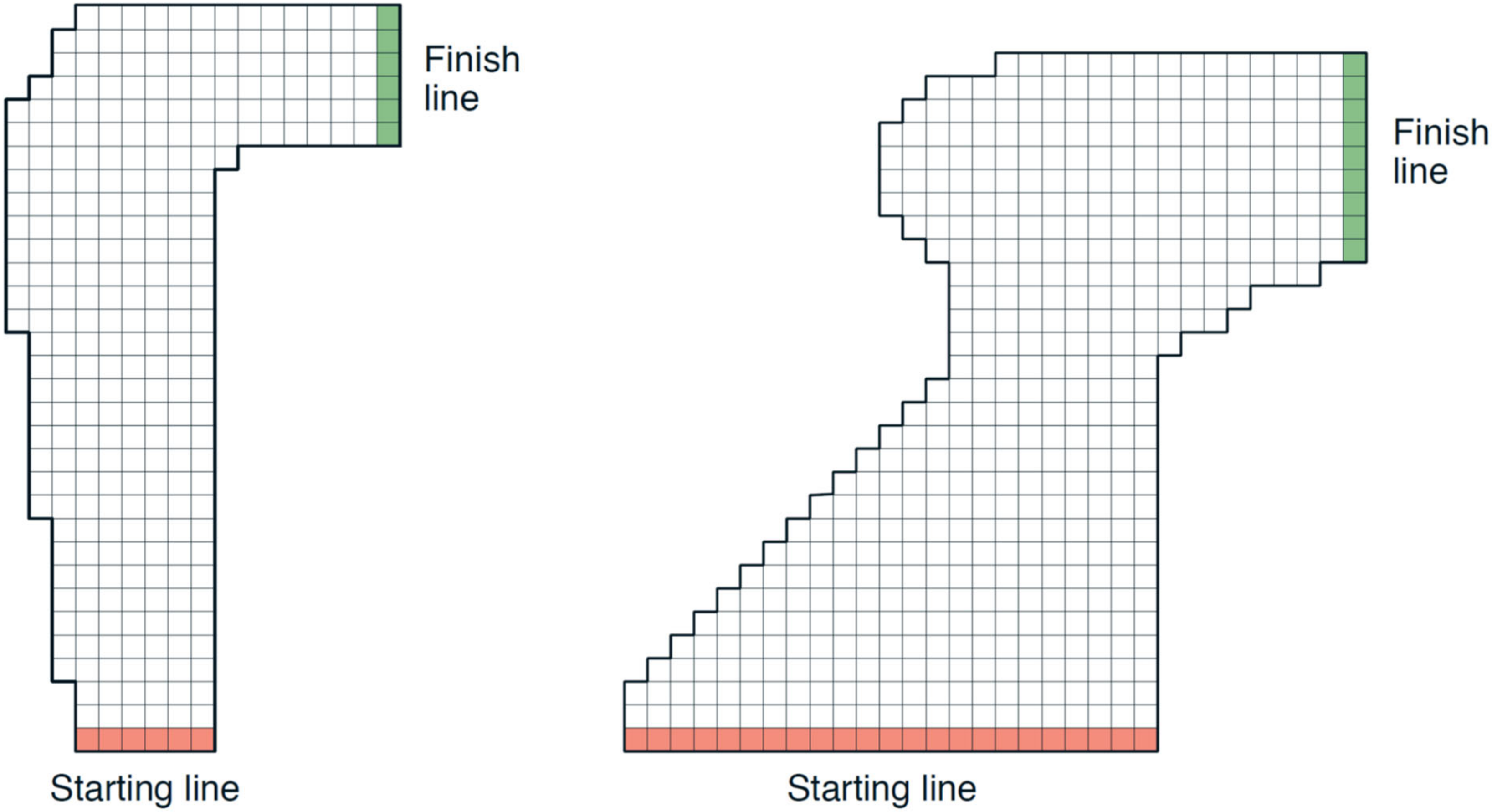


Figure 5.5: A couple of right turns for the racetrack task.

Submission

- Your implementation
- Around 10-15 slides with this structure
  - Main information for your implementation/experiments
  - Figures…
  - Discussion/Conclusions
- No report needed for this assignment
- Individual discussion of code with each group
- Submission and presentation either before
  - End of January (if you do NOT want to work in the holidays!) or
  - End of February

Discussion of assignment

- Discussion of code
- Implementation issues
- Methods