EL2805 Reinforcement Learning

Computer Lab 0

October 29, 2019

Division of Decision and Control Systems
School of Electrical Engineering and Computer Science
KTH Royal Institute of Technology

Key concepts: Markov Decision Processes, Dynamic Programming, Value Iteration.

Instructions: The two following problems are proposed to get familiar with python environment, and to apply some of the basic concepts of the course. This lab will further help in implementing mazes in an efficient manner. It is not graded. A solution (with code) will be published on canvas before Nov 14th.

1 Environment Setup

Throughout the course, we recommend the use of Python 3. To help you start we propose the following:

- Download and install Anaconda `https://www.anaconda.com/distribution/`.
- Create a folder with name `folder_name`.
- We recommend to set a conda environment:
  - Check documentation for more information `https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html#create-env-file-manually`
  - Download the file environment `rlenv.yml` from canvas and place it in the folder `folder_name`.
  - Create the conda environment by running the command: `conda env create -f rlenv.yml` and wait for the installation to finish.
  - If you are using Mac or Linux, run the environment by using the command: `source activate rl`. If you are using Windows, use the command: `activate rl` instead.
- If you decide not to create a conda environment, then you will need to install the requisite packages on your own using the command `pip` (see `https://docs.python.org/3/installing/index.html?highlight=pip`).
- The documentation for Python: `https://docs.python.org/3/`.
- For those new to Python or in need of a refresher, we recommend the following tutorial: `http://cs231n.github.io/python-numpy-tutorial/`.
- The solutions will be given in the form of Jupyter notebooks. Jupyter Notebook `https://jupyter.org/` is a web application for writing documents with live code. Check the website for installation.

Problem 1:
Shortest Path in the Maze

In this problem, consider the maze in Figure 1. The black cells represent obstacles or walls. The player enters the maze at A. By convention, the cells are labeled by their position on the x, y axis, so that $A = (0, 0)$ and $B = (5, 5)$. At each time step, the player can make a one-step move (up, down, right or left) or stay in her position. Her only objective, is to find the shortest path to the exit B, given full knowledge of the maze.

Note 1: The player cannot walk diagonally.

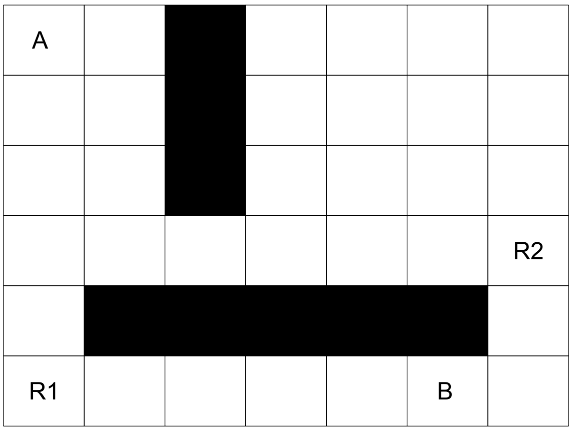

Figure 1: The maze.

(a) Formulate the problem as an MDP.

- Specify the state space $\mathcal{S}$.
- Specify the action space $\mathcal{A}$.
- Specify the transition probabilities $\mathcal{P}$.
- Specify the rewards $\mathcal{R}$.

(b) Solve the formulated MDP using dynamic programming, motivating your choice of the time horizon T and illustrate the shortest path solution, i.e., show the optimal action taken at each cell.

(c) Solve the formulated MDP using value iteration, and show the evolution of the values of each state.

(d) Solve the following modified problem. Every time the player visits the cell $R1$, she must stay there for 6 consecutive rounds with probability 0.5 (she can move as usual with probability 0.5). Similarly, when she visits $R2$, she must stay for 1 round with probability 0.5.

Hint: Modify the rewards only, so that they become random variables.

Problem 2:
Plucking berries

We consider a variation of Problem 1. Given the same maze as in Figure 1, the player now wants to find the most *rewarding* plucking path within T steps, starting at A, and given the following weight matrix W:

$$W = \begin{bmatrix} 0 & 1 & -\infty & 10 & 10 & 10 & 10 \\ 0 & 1 & -\infty & 10 & 0 & 0 & 10 \\ 0 & 1 & -\infty & 10 & 0 & 0 & 10 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & -\infty & -\infty & -\infty & -\infty & -\infty & 10 \\ 0 & 0 & 0 & 0 & 0 & 11 & 10 \end{bmatrix}.$$

The weight of each element is associated with the reward obtained upon moving on that cell.

(a) Modify your MDP formulation of Problem 1 so that it matches this setting.

(c) Solve the new MDP using dynamic programming, and illustrate the maximum value path. Do this for every T from 12 to 20.

(d) Solve the formulated MDP using value iteration, and show the obtained maximum value path.