Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institute for Dynamic Systems and Control
Institut für Dynamische Systeme und Regelungstechnik

151-0563-01 **Dynamic Programming and Optimal Control** (Fall 2018)

Programming Exercise Topic: Infinite Horizon Problems

Issued: Nov 22, 2018 Due: Dec 19, 2018

Rajan Gill(rgill@ethz.ch), Weixuan Zhang(wzhang@ethz.ch), November 21, 2018

Policy Iteration, Value Iteration, and Linear Programming

The goal of this programming exercise is to help a paparazzi take a picture of a celebrity in minimum time. For this purpose the paparazzi enters a celebrity's estate (see Fig. 1) and tries to sneak up to the celebrity's mansion in order to get a good picture. Unfortunately, there are various security cameras installed on the property. If caught on camera, the paparazzi will be brought to the entrance gate by the security guards and has to restart. In addition, there are various obstacles such as trees, bushes, ponds and pools on the property. The trees and bushes can be used by the paparazzi to hide from the cameras, but he cannot move through them. The ponds and pools can be crossed by the paparazzi, but this costs extra time and they do not offer any sight protection.

Figure 1: Bird's eye view of a celebrity's estate with trees, bushes, ponds and a pool.

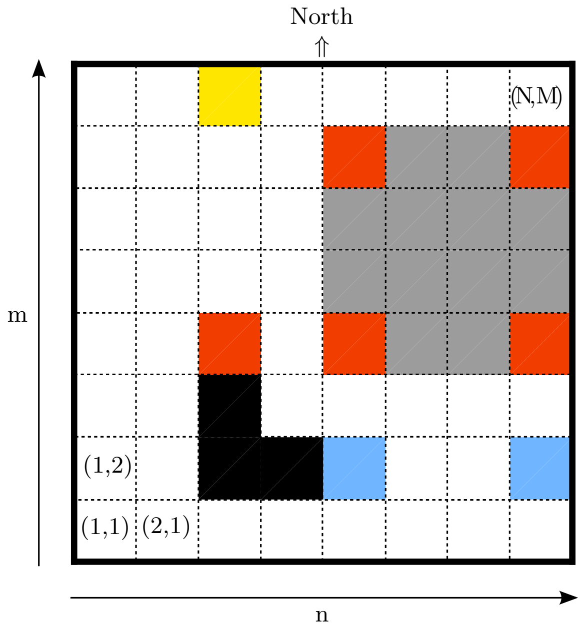

Figure 2: Example map of the celebrity's estate. The black cells represent trees and bushes, the blue cells ponds and pools, the grey cells the mansion, the red cells security cameras and the yellow cell the entrance gate.

Problem set up

The celebrity's property is discretized into $N \times M$ cells (see Fig. 2), where N is the width of the estate (from west to east) and M the length (from south to north), respectively. At each time step, the paparazzi can move north, west, south, east or stay at his current position and try to take a picture. The paparazzi's position is described by $x = (n, m)$, $n \in \{1, \dots, N\}$, $m \in \{1, \dots, M\}$.

The map of the estate is described by a $M \times N$-matrix, where positive values indicate cells that are inaccessible (e.g. trees, bushes, or the mansion) and negative values indicate ponds or pools. If the paparazzi decides to move into a pond or pool, it takes him four time steps. Leaving the pool or taking a picture from inside the pool only takes one time step. The positions of the mansion, which is always rectangular, is given by a $F \times 2$-matrix, where each row indicates a cell of the map that is part of the mansion.

The positions of the security cameras are given by a $H \times 3$-matrix, where each row indicates the cell of the camera position (first two columns) and the camera's image quality (third column). The cameras can film in all four cardinal directions, but not diagonal or through trees, bushes or the mansion. If the paparazzi moves to a cell that is in line of sight of a camera, the probability of being detected by this camera is $\gamma_i/||d_{pc}||$, where γ_i denotes the camera's image quality and $||d_{pc}||$ the current distance between the paparazzi and the camera (measured in number of cells). In case the paparazzi is caught on camera, he will be brought to the entrance gate which costs an additional six time steps. The position of the entrance gate is given by a 1×2-matrix indicating the cell of the map where the gate is located.

In each time step, the paparazzi can take a picture instead of moving around. The minimum probability of successfully taking picture is $p_c = 0.001$, which represents the rare case when the celebrity walks into the picture outside of the mansion. If any cell of the mansion is in the field of view of the paparazzi (all four cardinal directions, not diagonal or through trees or bushes), the probability of successfully taking a picture is $\max\{p_c, \gamma_p/||d_{pm}||\}$, with $\gamma_p = 0.5$ and $||d_{pm}||$ being the current distance between the paparazzi and the corresponding cell of the mansion.

Note: The order of events during a time step is the following: First, the paparazzi moves to a new cell or takes a picture. If he successfully took a picture of the celebrity, the task is over. In all other cases, the security cameras try to spot the paparazzi and if caught on camera, the paparazzi is escorted to entrance gate by the security guards.

If the paparazzi moves into a pond or pool cell, the security cameras have four attempts to film the paparazzi and hence the probability of the paparazzi being detected increases.

Tasks

Find the policy minimizing the expected number of time steps required to successfully take a picture by

a) creating a transition probability matrix $P \in \mathbb{R}^{K \times K \times L}$, where K is the number of possible states and L is the number of control inputs[1]. For creating P, the state space is created by assigning a unique state index $i = 1, 2, \ldots, K$ to all accessible cells of the estate (see `main.m`).
Use the provided file `ComputeTransitionProbabilities.m` as a template for your implementation.
This part counts 30% towards the grade.

b) creating a stage cost matrix [2] $G \in \mathbb{R}^{K \times L}$.
Use the provided file `ComputeStageCosts.m` as a template for your implementation.
This part counts 25% towards the grade.

c) applying Value Iteration[3], Policy Iteration and Linear Programming[4] to compute $J \in \mathbb{R}^K$ and the optimal policy $\mu(i),\ i = 1, \ldots, K$, that solves the stochastic shortest path problem.
Use the provided files `ValueIteration.m`, `PolicyIteration.m` and `LinearProgramming.m` as a template for your implementation.
Each algorithm makes up for 15% of the grade.

---

[1] Set the transition probabilities to 0 for infeasible moves.

[2] Set the expected stage cost to `inf` for infeasible moves.

[3] You can terminate the algorithm if all $J(i),\ i = 1, \ldots, K$, do not change by more than 10^{-5} within one iteration step.

[4] In your implementation of the file LinearProgramming.m, you may use the MATLAB function "linprog" to solve the linear program.

Provided Matlab files

A set of MATLAB files is provided on the class website. Use them for solving the above problem. Strictly follow the structure. Grading is automated using MATLAB 2018b. You can add functions to the template files, but each file should be self-contained, i.e. not depend on any external custom function.

`main.m`	Matlab script that has to be used to generate a map of the estate, execute the stochastic shortest path algorithms and display the results.
`GenerateMap.p`	Matlab function that generates a map of the celebrity's estate.
`PlotMap.m`	Matlab function that can plot a map of the estate, and the cost and control action for each accessible cell.
`PlotMap3.m`	Matlab function that can plot a three dimensional view of the estate and control action for each accessible cell.
`ComputeTransitionProbabilities.m`	Matlab function template to be used for creating the transition probability matrix $P \in \mathbb{R}^{K \times K \times L}$.
`ComputeStageCosts.m`	Matlab function template to be used for creating the stage cost matrix $G \in \mathbb{R}^{K \times L}$.
`ValueIteration.m`	Matlab function template to be used for your implementation of the Value Iteration algorithm for the stochastic shortest path problem.
`PolicyIteration.m`	Matlab function template to be used for your implementation of the Policy Iteration algorithm for the stochastic shortest path problem.
`LinearProgramming.m`	Matlab function template to be used for your implementation of the Linear Programming algorithm for the stochastic shortest path problem.
`exampleMap.mat`	A pre-generated 15×10 map to be used for testing your implementations of the above functions.
`exampleP.mat`	The transition probability matrix $P \in \mathbb{R}^{K \times K \times L}$ for the example map.
`exampleG.mat`	The stage cost matrix $G \in \mathbb{R}^{K \times L}$ for the example map.

Deliverables

Please hand in by e-mail

- your MATLAB implementation of the following files:
  ComputeTransitionProbabilites.m,
  ComputeStageCost.m,
  ValueIteration.m,
  PolicyIteration.m,
  LinearProgramming.m.
  Only submit the above mentioned files. Your code should not depend on any other non-standard MATLAB functions.

- in a PDF-file a scanned declaration of originality, signed by each student to confirm that the work is original and has been done by the author(s) independently:
  https://www.ethz.ch/content/dam/ethz/main/education/rechtliches-abschluesse/leistungskontrollen/declaration-originality.pdf.
  Each work submitted will be tested for plagiarism.

Please include all files into one zip-archive, named DPOCEx_Names.zip, where Names is a list of the full names of all students who have worked on the solution.
(e.g DPOCEx_GillRajan_ZhangWeixuan.zip)[5]

Send your files to wzhang@ethz.ch with subject [programming exercise submission] by the due date indicated above. We will send a confirmation e-mail upon receiving your e-mail. You are ultimately responsible that we receive your solution in time.

[5]Up to three students are allowed to work together on the problem. They will all receive the same grade.