ECE59500RL HW2

Robert (Cars) Chandler — chandl71@purdue.edu

Problem 1

Problem 2

Problem 3

Problem 4

First, we need to assign labels to the states, which are the different spaces on the board. We will use a zero-indexed x-y coordinate system to refer to the different states $s_{xy} \in \mathcal{S}$, with the origin at the bottom left square, $s_{0,0}$. Moving horizontally will increase the x-component and vertically the y-component, so that our state space is

$$\mathcal{S} = \{s_{ij} : i, j \in \mathbb{N}_0, \quad i, j \leq 5\}$$

> **Note**
>
> Although we are using two "dimensions" to identify each state, we still treat it as a one-dimensional vector, so that we have one row for each $s \in \mathcal{S}$ in $\vec{v}$, P^π, and so forth. The order of the states for this vector will always be in row-major order:
>
> $$(0,0), (1,0), (2,0), (3,0), (4,0), (0,1), (1,1), ..., (3,4), (4,4)$$

4.1.a

The policy can be evaluated analytically using the following equation:

$$\vec{v}^\pi = (I - \gamma P^\pi)^{-1} \vec{R}^\pi$$

We have $\gamma = 0.95$ from the problem statement. P^π and $\vec{R}^\pi$ each need to be evaluated by going over each state in $\mathcal{S}$ and using the information given to us to evaluate them. Beginning with P^π:

$$P^\pi_{ij} = P(s_j|s_i, a) = P(s_j|s_i, \pi(s_i))$$

We can use Python to encode the logic described in the problem statement to programatically calculate P^π for each state transition:

```
from enum import Enum
import numpy as np
import itertools
from IPython.display import Markdown


class Space(Enum):
    LIGHTNING = -1
    NORMAL = 0
    MOUNTAIN = 1
    TREASURE = 2


width = 5

board = np.full([width, width], Space.NORMAL)

board[2, 1] = Space.MOUNTAIN
board[3, 1] = Space.MOUNTAIN
board[1, 3] = Space.MOUNTAIN
board[2, 3] = Space.LIGHTNING
board[4, 4] = Space.TREASURE

policy = np.array(
    [
        list("URRUU"),
        list("UDDDU"),
        list("UURRR"),
        list("LULLU"),
        list("RRRRU"),
    ]
).T
```

```python
p = np.zeros([25, 25])


def i_1d(x, y):
    return np.ravel_multi_index([y, x], dims=[width, width])


def is_blocked(x, y):
    return x < 0 or y < 0 or x >= width or y >= width or board[x, y] == Space.MOUNTAIN


for x, y in itertools.product(range(5), range(5)):
    i_cur_1d = i_1d(x, y)

    if board[x, y] != Space.NORMAL:
        p[i_cur_1d, i_cur_1d] = 1
        continue

    for a, (i2, j2) in zip(
        list("LRUD"), [[x - 1, y], [x + 1, y], [x, y + 1], [x, y - 1]]
    ):
        prob = 0.85 if a == policy[x, y] else 0.05
        if is_blocked(i2, j2):
            p[i_cur_1d, i_cur_1d] += prob
        else:
            p[i_cur_1d, i_1d(i2, j2)] += prob

state_text = []
for i in range(width**2):
    y1, x1 = np.unravel_index(i, [width, width])
    a = policy[x1, y1]
    for j in np.argwhere(p[i]).flatten():
        y2, x2 = np.unravel_index(j, [width, width])
        state_text.append(
            rf"P^{{\pi}}(s_{{ {x2}, {y2} }} | s_{{ {x1}, {y1} }}, \pi(s_{{ {x1},"
            rf" {y1} }}) = \text{{{a}}}) &= {p[i, j]:g} \\"
        )
    state_text.append(r"\\")

state_text = "\n".join(state_text)
```

The resulting state transition probabilities are listed as follows:

$$P^\pi(s_{0,0}|s_{0,0},\pi(s_{0,0})=\mathrm{U})=0.1$$
$$P^\pi(s_{1,0}|s_{0,0},\pi(s_{0,0})=\mathrm{U})=0.05$$
$$P^\pi(s_{0,1}|s_{0,0},\pi(s_{0,0})=\mathrm{U})=0.85$$

$$P^\pi(s_{0,0}|s_{1,0},\pi(s_{1,0})=\mathrm{R})=0.05$$
$$P^\pi(s_{1,0}|s_{1,0},\pi(s_{1,0})=\mathrm{R})=0.05$$
$$P^\pi(s_{2,0}|s_{1,0},\pi(s_{1,0})=\mathrm{R})=0.85$$
$$P^\pi(s_{1,1}|s_{1,0},\pi(s_{1,0})=\mathrm{R})=0.05$$

$$P^\pi(s_{1,0}|s_{2,0},\pi(s_{2,0})=\mathrm{R})=0.05$$
$$P^\pi(s_{2,0}|s_{2,0},\pi(s_{2,0})=\mathrm{R})=0.1$$
$$P^\pi(s_{3,0}|s_{2,0},\pi(s_{2,0})=\mathrm{R})=0.85$$

$$P^\pi(s_{2,0}|s_{3,0},\pi(s_{3,0})=\mathrm{U})=0.05$$
$$P^\pi(s_{3,0}|s_{3,0},\pi(s_{3,0})=\mathrm{U})=0.9$$
$$P^\pi(s_{4,0}|s_{3,0},\pi(s_{3,0})=\mathrm{U})=0.05$$

$$P^\pi(s_{3,0}|s_{4,0},\pi(s_{4,0})=\mathrm{U})=0.05$$
$$P^\pi(s_{4,0}|s_{4,0},\pi(s_{4,0})=\mathrm{U})=0.1$$
$$P^\pi(s_{4,1}|s_{4,0},\pi(s_{4,0})=\mathrm{U})=0.85$$

$$P^\pi(s_{0,0}|s_{0,1},\pi(s_{0,1})=\mathrm{U})=0.05$$
$$P^\pi(s_{0,1}|s_{0,1},\pi(s_{0,1})=\mathrm{U})=0.05$$
$$P^\pi(s_{1,1}|s_{0,1},\pi(s_{0,1})=\mathrm{U})=0.05$$
$$P^\pi(s_{0,2}|s_{0,1},\pi(s_{0,1})=\mathrm{U})=0.85$$

$$P^\pi(s_{1,0}|s_{1,1},\pi(s_{1,1})=\mathrm{D})=0.85$$
$$P^\pi(s_{0,1}|s_{1,1},\pi(s_{1,1})=\mathrm{D})=0.05$$
$$P^\pi(s_{1,1}|s_{1,1},\pi(s_{1,1})=\mathrm{D})=0.05$$
$$P^\pi(s_{1,2}|s_{1,1},\pi(s_{1,1})=\mathrm{D})=0.05$$

$$P^\pi(s_{2,1}|s_{2,1},\pi(s_{2,1})=\mathrm{D})=1$$

$$P^{\pi}(s_{3,1}|s_{3,1},\pi(s_{3,1})=\mathrm{D})=1$$

$$P^{\pi}(s_{4,0}|s_{4,1},\pi(s_{4,1})=\mathrm{U})=0.05$$
$$P^{\pi}(s_{4,1}|s_{4,1},\pi(s_{4,1})=\mathrm{U})=0.1$$
$$P^{\pi}(s_{4,2}|s_{4,1},\pi(s_{4,1})=\mathrm{U})=0.85$$

$$P^{\pi}(s_{0,1}|s_{0,2},\pi(s_{0,2})=\mathrm{U})=0.05$$
$$P^{\pi}(s_{0,2}|s_{0,2},\pi(s_{0,2})=\mathrm{U})=0.05$$
$$P^{\pi}(s_{1,2}|s_{0,2},\pi(s_{0,2})=\mathrm{U})=0.05$$
$$P^{\pi}(s_{0,3}|s_{0,2},\pi(s_{0,2})=\mathrm{U})=0.85$$

$$P^{\pi}(s_{1,1}|s_{1,2},\pi(s_{1,2})=\mathrm{U})=0.05$$
$$P^{\pi}(s_{0,2}|s_{1,2},\pi(s_{1,2})=\mathrm{U})=0.05$$
$$P^{\pi}(s_{1,2}|s_{1,2},\pi(s_{1,2})=\mathrm{U})=0.85$$
$$P^{\pi}(s_{2,2}|s_{1,2},\pi(s_{1,2})=\mathrm{U})=0.05$$

$$P^{\pi}(s_{1,2}|s_{2,2},\pi(s_{2,2})=\mathrm{R})=0.05$$
$$P^{\pi}(s_{2,2}|s_{2,2},\pi(s_{2,2})=\mathrm{R})=0.05$$
$$P^{\pi}(s_{3,2}|s_{2,2},\pi(s_{2,2})=\mathrm{R})=0.85$$
$$P^{\pi}(s_{2,3}|s_{2,2},\pi(s_{2,2})=\mathrm{R})=0.05$$

$$P^{\pi}(s_{2,2}|s_{3,2},\pi(s_{3,2})=\mathrm{R})=0.05$$
$$P^{\pi}(s_{3,2}|s_{3,2},\pi(s_{3,2})=\mathrm{R})=0.05$$
$$P^{\pi}(s_{4,2}|s_{3,2},\pi(s_{3,2})=\mathrm{R})=0.85$$
$$P^{\pi}(s_{3,3}|s_{3,2},\pi(s_{3,2})=\mathrm{R})=0.05$$

$$P^{\pi}(s_{4,1}|s_{4,2},\pi(s_{4,2})=\mathrm{R})=0.05$$
$$P^{\pi}(s_{3,2}|s_{4,2},\pi(s_{4,2})=\mathrm{R})=0.05$$
$$P^{\pi}(s_{4,2}|s_{4,2},\pi(s_{4,2})=\mathrm{R})=0.85$$
$$P^{\pi}(s_{4,3}|s_{4,2},\pi(s_{4,2})=\mathrm{R})=0.05$$

$$P^{\pi}(s_{0,2}|s_{0,3},\pi(s_{0,3})=\mathrm{L})=0.05$$

$$P^\pi(s_{0,3}|s_{0,3}, \pi(s_{0,3}) = \mathrm{L}) = 0.9$$
$$P^\pi(s_{0,4}|s_{0,3}, \pi(s_{0,3}) = \mathrm{L}) = 0.05$$

$$P^\pi(s_{1,3}|s_{1,3}, \pi(s_{1,3}) = \mathrm{U}) = 1$$

$$P^\pi(s_{2,3}|s_{2,3}, \pi(s_{2,3}) = \mathrm{L}) = 1$$

$$P^\pi(s_{3,2}|s_{3,3}, \pi(s_{3,3}) = \mathrm{L}) = 0.05$$
$$P^\pi(s_{2,3}|s_{3,3}, \pi(s_{3,3}) = \mathrm{L}) = 0.85$$
$$P^\pi(s_{4,3}|s_{3,3}, \pi(s_{3,3}) = \mathrm{L}) = 0.05$$
$$P^\pi(s_{3,4}|s_{3,3}, \pi(s_{3,3}) = \mathrm{L}) = 0.05$$

$$P^\pi(s_{4,2}|s_{4,3}, \pi(s_{4,3}) = \mathrm{U}) = 0.05$$
$$P^\pi(s_{3,3}|s_{4,3}, \pi(s_{4,3}) = \mathrm{U}) = 0.05$$
$$P^\pi(s_{4,3}|s_{4,3}, \pi(s_{4,3}) = \mathrm{U}) = 0.05$$
$$P^\pi(s_{4,4}|s_{4,3}, \pi(s_{4,3}) = \mathrm{U}) = 0.85$$

$$P^\pi(s_{0,3}|s_{0,4}, \pi(s_{0,4}) = \mathrm{R}) = 0.05$$
$$P^\pi(s_{0,4}|s_{0,4}, \pi(s_{0,4}) = \mathrm{R}) = 0.1$$
$$P^\pi(s_{1,4}|s_{0,4}, \pi(s_{0,4}) = \mathrm{R}) = 0.85$$

$$P^\pi(s_{0,4}|s_{1,4}, \pi(s_{1,4}) = \mathrm{R}) = 0.05$$
$$P^\pi(s_{1,4}|s_{1,4}, \pi(s_{1,4}) = \mathrm{R}) = 0.1$$
$$P^\pi(s_{2,4}|s_{1,4}, \pi(s_{1,4}) = \mathrm{R}) = 0.85$$

$$P^\pi(s_{2,3}|s_{2,4}, \pi(s_{2,4}) = \mathrm{R}) = 0.05$$
$$P^\pi(s_{1,4}|s_{2,4}, \pi(s_{2,4}) = \mathrm{R}) = 0.05$$
$$P^\pi(s_{2,4}|s_{2,4}, \pi(s_{2,4}) = \mathrm{R}) = 0.05$$
$$P^\pi(s_{3,4}|s_{2,4}, \pi(s_{2,4}) = \mathrm{R}) = 0.85$$

$$P^\pi(s_{3,3}|s_{3,4}, \pi(s_{3,4}) = \mathrm{R}) = 0.05$$
$$P^\pi(s_{2,4}|s_{3,4}, \pi(s_{3,4}) = \mathrm{R}) = 0.05$$
$$P^\pi(s_{3,4}|s_{3,4}, \pi(s_{3,4}) = \mathrm{R}) = 0.05$$

$$P^\pi(s_{4,4}|s_{3,4}, \pi(s_{3,4}) = \text{R}) = 0.85$$

$$P^\pi(s_{4,4}|s_{4,4}, \pi(s_{4,4}) = \text{U}) = 1$$

All other possible state transitions have probability 0.

Moving onto $\vec{R}^\pi$:

$$\vec{R}^\pi = \begin{bmatrix} R(s_{1,1}, \pi(s_{1,1}) \\ R(s_{1,2}, \pi(s_{1,2}) \\ \cdots \\ R(s_{4,4}, \pi(s_{4,4})) \end{bmatrix}_{|\mathcal{S}| \times 1}$$

So given the reward function described, we just have a simple vector with two nonzero elements:

```
r = np.zeros(width * width, dtype=int)
r[i_1d(*np.argwhere(board == Space.LIGHTNING).squeeze())] = -1
r[i_1d(*np.argwhere(board == Space.TREASURE).squeeze())] = 1
```

$$\vec{R}^\pi = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 1]^T$$