

# Planning, Learning and Decision Making

Group 27

78375 - João Pirralha

84758 - Rafael Ribeiro

## Homework 3. Partially observable Markov decision problems

1.

$$\mathcal{X} = \{A, B_1, B_2, C, D, E, F\}$$

$$\mathcal{A} = \{a, b, c\}$$

$$\mathcal{O} = \{A, B, C, D, E, F\}$$

2.

```

In [1]: import numpy as np

# States
A = 0; B1 = 1; B2 = 2; C = 3; D = 4; E = 5; F = 6

Pa = np.zeros((7, 7))
Pa[A, B1] = 0.5
Pa[A, B2] = 0.5
Pa[B1, E] = 1
Pa[B2, F] = 1
Pa[C, B1] = 1
Pa[D, B2] = 1
Pa[E, A] = 1
Pa[F, A] = 1

Pb = np.zeros((7, 7))
Pb[A, B1] = 0.5
Pb[A, B2] = 0.5
Pb[B1, F] = 1
Pb[B2, E] = 1
Pb[C, B1] = 1
Pb[D, B2] = 1
Pb[E, A] = 1
Pb[F, A] = 1

Pc = np.zeros((7, 7))
Pc[A, B1] = 0.5
Pc[A, B2] = 0.5
Pc[B1, C] = 1
Pc[B2, D] = 1
Pc[C, B1] = 1
Pc[D, B2] = 1
Pc[E, A] = 1
Pc[F, A] = 1

# Observations
ZA = 0; ZB = 1; ZC = 2; ZD = 3; ZE = 4; ZF = 5
Oa = np.zeros((7, 6))
Oa[A, ZA] = 1
Oa[B1, ZB] = 1
Oa[B2, ZB] = 1
Oa[C, ZC] = 1
Oa[D, ZD] = 1
Oa[E, ZE] = 1
Oa[F, ZF] = 1
Oc = Ob = Oa

# Actions
a = 0; b = 1; c = 2
C = np.ones((7, 3))
C[F, (a, b, c)] = 0

print("Pa:")
print(Pa)
print("\nPb:")
print(Pb)
print("\nPc:")
print(Pc)
print("\nOa = Ob = Oc:")
print(Oa)
print("\nC:")
print(C)

```

```

Pa:
[[0.  0.5 0.5 0.  0.  0.  0. ]
 [0.  0.  0.  0.  0.  1.  0. ]
 [0.  0.  0.  0.  0.  0.  1. ]
 [0.  1.  0.  0.  0.  0.  0. ]
 [0.  0.  1.  0.  0.  0.  0. ]
 [1.  0.  0.  0.  0.  0.  0. ]
 [1.  0.  0.  0.  0.  0.  0. ]]

```

```

Pb:
[[0.  0.5 0.5 0.  0.  0.  0. ]
 [0.  0.  0.  0.  0.  0.  1. ]
 [0.  0.  0.  0.  0.  1.  0. ]
 [0.  1.  0.  0.  0.  0.  0. ]
 [0.  0.  1.  0.  0.  0.  0. ]
 [1.  0.  0.  0.  0.  0.  0. ]
 [1.  0.  0.  0.  0.  0.  0. ]]

```

```

Pc:
[[0.  0.5 0.5 0.  0.  0.  0. ]
 [0.  0.  0.  1.  0.  0.  0. ]
 [0.  0.  0.  0.  1.  0.  0. ]
 [0.  1.  0.  0.  0.  0.  0. ]
 [0.  0.  1.  0.  0.  0.  0. ]
 [1.  0.  0.  0.  0.  0.  0. ]
 [1.  0.  0.  0.  0.  0.  0. ]]

```

```

0a = 0b = 0c:
[[1. 0. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0. 0.]
 [0. 0. 1. 0. 0. 0.]
 [0. 0. 0. 1. 0. 0.]
 [0. 0. 0. 0. 1. 0.]
 [0. 0. 0. 0. 0. 1.]]

```

```

C:
[[1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]
 [0. 0. 0.]]

```

3.

```

In [2]: bt = np.array([0, 0.5, 0.5, 0, 0, 0, 0])
print("The agent selects action a at time step t:")
bta = bt @ Pa
print(bta)
print("The agent selects action b at time step t:")
btb = bt @ Pb
print(btb)
print("The agent selects action c at time step t:")
btc = bt @ Pc
print(btc)

```

```

The agent selects action a at time step t:
[0.  0.  0.  0.  0.  0.5 0.5]
The agent selects action b at time step t:
[0.  0.  0.  0.  0.  0.5 0.5]
The agent selects action c at time step t:
[0.  0.  0.  0.5 0.5 0.  0. ]

```