Optimal decision making for complex problems : First Assignment

Libert Robin 20186942

Otmani Chabane Guillaume 20163387

11 février 2020

Sectrion 2 : Implementation Of The Domain

For this simulation, we have decided to implement the stationary policy "always go right", with 10 iterations (because once the agent is on the border of the domain and tries to go to an inaccessible position, it just stays at the same state).

Here is the trajectory of the agent for both domain :

Step	Deterministic Domain	Stochastic Domain
0	(3,0)	(3,0)
1	(3,1)	(3,1)
2	(3,2)	(3,2)
3	(3,3)	(0,0)
4	(3,4)	(0,1)
5	(3,4)	(0,0)
6	(3,4)	(0,0)
7	(3,4)	(0,0)
8	(3,4)	(0,1)
9	(3,4)	(0,0)
10	(3,4)	(0,1)

Section 3 : Expected Return Of A Policy

We can estimate the value of J^{μ}, with the expression of the maximum error committed at the step N :

$$\frac{\gamma^N}{1-\gamma} B_r$$

With B_r the maximum reward we can get on the define domain. In the domain of this project, it corresponds to the value 19.
Here, we decided to work with a maximum absolute error of 0.5, which means to 820 iterations of the recurrence equation of J_N^{μ} in a deterministic domain :

$$J_N^{\mu}(x) = r(x, \mu(x)) \ + \ \gamma \, J_{N-1}^{\mu}(\ f(x, \mu(x)) \ )$$

And in a stochastic domain :

$$J_N^{\mu}(x) = \underset{w \sim P_w(\cdot|x,u)}{E} [\ r(x, \mu(x), w) \ + \ \gamma \, J_{N-1}^{\mu}(\ f(x, \mu(x), w) \ ) \ ]$$

Here are some tabulars with the corresponding function J_N^{μ}, for each cell of the domain, with 820 iterations and an "always go right" policy, for the deterministic domain :

Cell	J_N^{μ}
(0,0)	1839,1
(0,1)	1856,6
(0,2)	1880,4
(0,3)	1899,4
(0,4)	1899,4

Cell	J_N^{μ}
(1,0)	989,7
(1,1)	996,7
(1,2)	998,7
(1,3)	999,7
(1,4)	999,7

Cell	J_N^{μ}
(2,0)	-779,0
(2,1)	-778,8
(2,2)	-790,7
(2,3)	-799,7
(2,4)	-799,7

Cell	J_N^{μ}
(3,0)	-471,4
(3,1)	-467,1
(3,2)	-475,8
(3,3)	-499,8
(3,4)	-499,8

Cell	J_N^{μ}
(4,0)	849,1
(4,1)	874,8
(4,2)	887.7
(4,3)	899,7
(4,4)	899,7

Same data but for a stochastic domain :

Cell	J_N^μ
(0,0)	-70,6
(0,1)	-102,3
(0,2)	-1,5
(0,3)	-137,3
(0,4)	-89,0

Cell	J_N^μ
(1,0)	-66,2
(1,1)	-91,1
(1,2)	-126,6
(1,3)	-114,5
(1,4)	-128,7

Cell	J_N^μ
(2,0)	-104,4
(2,1)	-52,6
(2,2)	-52,3
(2,3)	-5,8
(2,4)	1,4

Cell	J_N^μ
(3,0)	-79,9
(3,1)	-33,0
(3,2)	-142,0
(3,3)	-102,1
(3,4)	-2,2

Cell	J_N^μ
(4,0)	-69,4
(4,1)	-56,6
(4,2)	-81,8
(4,3)	-63,8
(4,4)	-104,7

Section 4 : Optimal Policy

To compute $p(x' \mid x, u)$ and $r(x, u)$, we have apply to the agent a routine of T steps, with always a random uniform policy.
Then, with the estimation of $p(x' \mid x, u)$ and $r(x, u)$ and the Q_N function computed for each cells with our exploration of the domain in T steps, we can defined an optimal policy and its expected return.
Here is a tabular for the optimal policy, more exactly the appropriate move to execute when we are in each cell :

Cell	Move
(0,0)	(1,0)
(0,1)	(0,1)
(0,2)	(0,1)
(0,3)	(0,1)
(0,4)	(-1,0)

Cell	move
(1,0)	(0,1)
(1,1)	(0,1)
(1,2)	(0,1)
(1,3)	(0,1)
(1,4)	(-1,0)

Cell	Move
(2,0)	(-1,0)
(2,1)	(0,1)
(2,2)	(-1,0)
(2,3)	(-1,0)
(2,4)	(-1,0)

Cell	Move
(3,0)	(-1,0)
(3,1)	(0,1)
(3,2)	(0,1)
(3,3)	(-1,0)
(3,4)	(-1,0)

Cell	Move
(4,0)	(-1,0)
(4,1)	(0,1)
(4,2)	(-1,0)
(4,3)	(-1,0)
(4,4)	(0,-1)

We did not arrive to find the smallest value of T, but we can observe that for a huge T, J_N converges to there value, for each cell (in the deterministic domain and with the optimal policy).

Cell	J_N^μ
(0,0)	1842,0
(0,1)	1857,1
(0,2)	1880,9
(0,3)	1899,9
(0,4)	1899,9

Cell	J_N^μ
(1,0)	1854,5
(1,1)	1870,2
(1,2)	1881,0
(1,3)	1880,9
(1,4)	1899,9

Cell	J_N^μ
(2,0)	1842,0
(2,1)	1855,7
(2,2)	1870,2
(2,3)	1881,0
(2,4)	1890,9

Cell	J_N^μ
(3,0)	1828,6
(3,1)	1849,0
(3,2)	1863,6
(3,3)	1863,2
(3,4)	1864,0

Cell	J_N^μ
(4,0)	1816,3
(4,1)	1826,5
(4,2)	1849,0
(4,3)	1863,6
(4,4)	1842,0