UNIVERSITÉ LIBRE DE BRUXELLES ULB

Learning Dynamics

INFO-F409

Assignment 3

19 December 2021

TAILLANDY Valentin

ULB student ID : 000542194

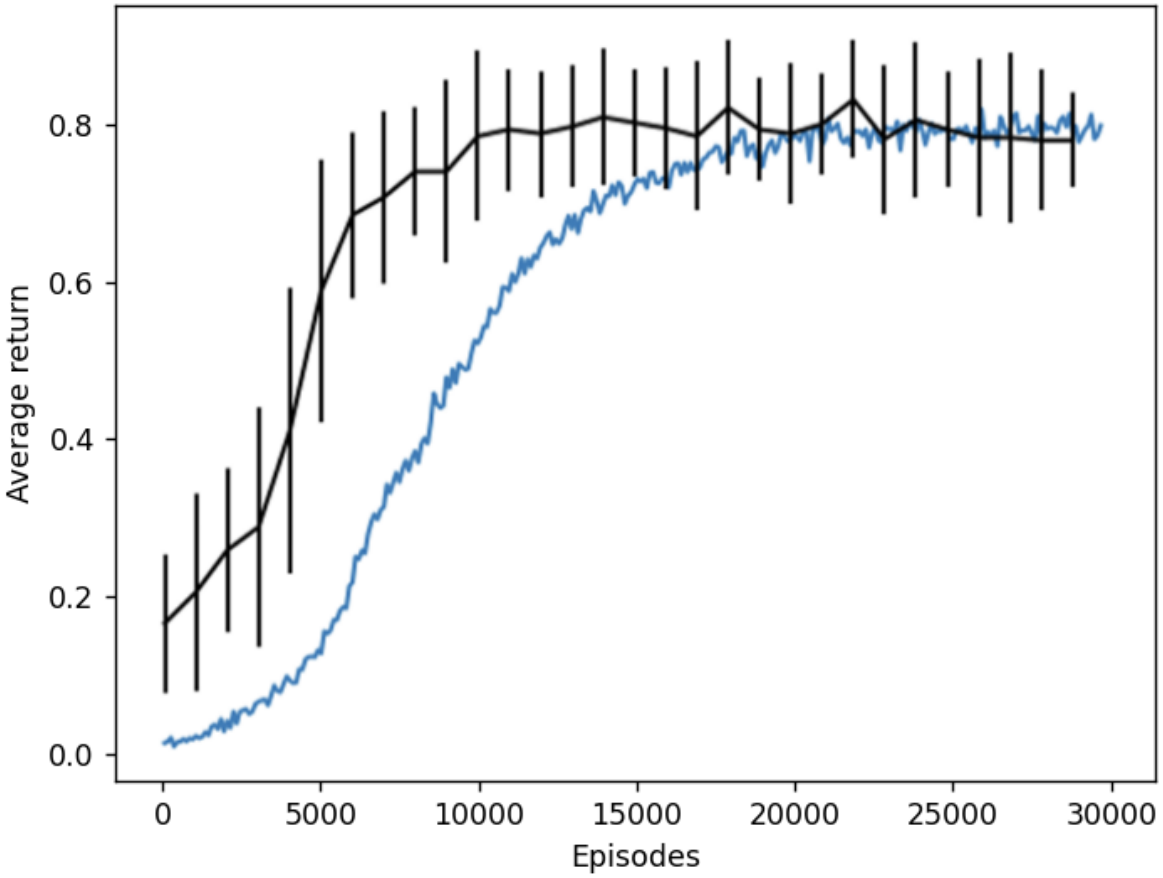
UNIVERSITÉ **L**IBRE DE **B**RUXELLES (ULB)

Contents

Single-agent RL: Frozen Lake

Parameters:

- Discount factor (γ) = 1.
- Learning rate = 0.0165
- Epsilon max = 1.
- Epsilon min = 0.
- Decay rate = 0.99974



The blue represents the average returns per episode. The black part represents the return of the agent during evaluation with the standard deviation as error bars.

Policies learned by the agent:

```
[['Left' 'Up' 'Up' 'Up']
 ['Left' 'Hole' 'Right' 'Hole']
 ['Up' 'Down' 'Left' 'Hole']
 ['Hole' 'Right' 'Down' 'End']]
```

```
[['Left' 'Up' 'Left' 'Down']
 ['Left' 'Hole' 'Right' 'Hole']
 ['Up' 'Down' 'Left' 'Hole']
 ['Hole' 'Right' 'Down' 'End']]
```

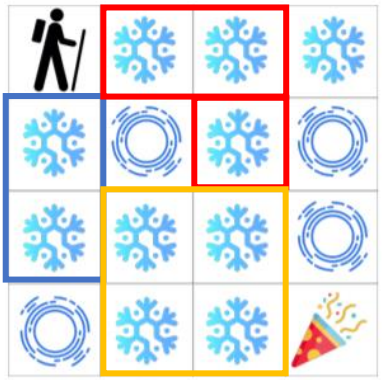
```
[['Left' 'Up' 'Left' 'Up']
 ['Left' 'Hole' 'Left' 'Hole']
 ['Up' 'Down' 'Left' 'Hole']
 ['Hole' 'Right' 'Down' 'End']]
```

```
[['Left' 'Up' 'Left' 'Left']
 ['Left' 'Hole' 'Left' 'Hole']
 ['Up' 'Down' 'Left' 'Hole']
 ['Hole' 'Right' 'Down' 'End']]
```

Most common policy:

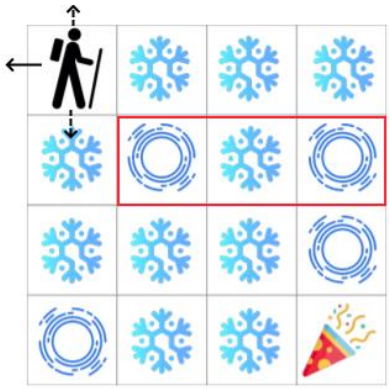
```
[['Left' 'Up' 'Up' 'Up']
 ['Left' 'Hole' 'Right' 'Hole']
 ['Up' 'Down' 'Left' 'Hole']
 ['Hole' 'Right' 'Down' 'End']]
```

Most policies are variant of this very policy. A preferred path is traced, and one is as much as possible avoided, hence the variations mostly come from this path (red):



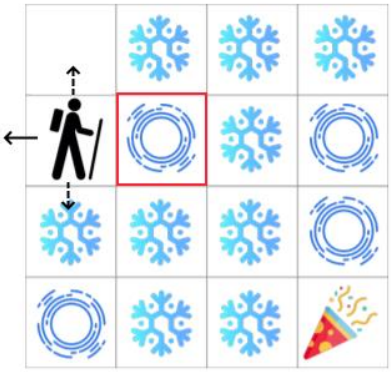
(Blue preferred path, red avoided path, orange common path)

The first move that is chosen by the agent is a move made to avoid the red path with the two holes facing each-other. The agent has learned that he cannot go backwards, so he will use this knowledge to avoid the unwanted path by going in the opposite direction of the unwanted path and the wanted path will be gambled:

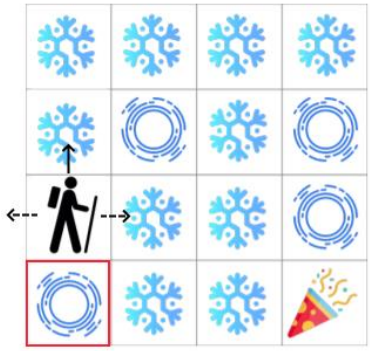


(Arrows with dashes are the possible outcomes while full arrows are the chosen direction)

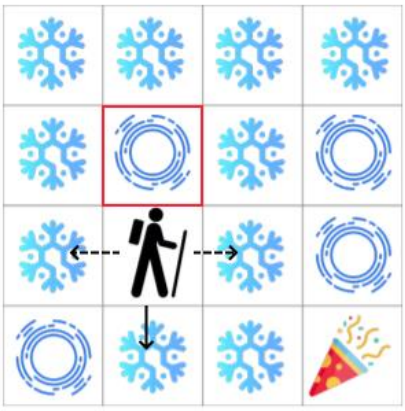
By going left first the agent has a probability of 66% to not move and 33% to go down. If he goes to the right, the agent will need to go towards the two facing holes and will have at minima 33% chance of falling into one and at maxima 66% chance of falling into one, therefore he must avoid this path.



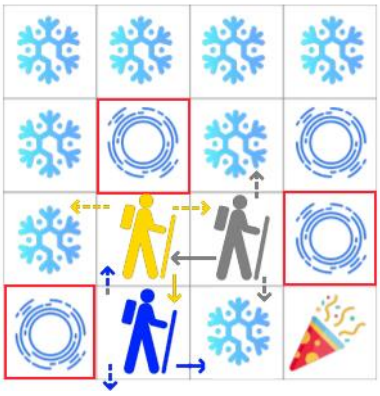
Same thing here, if the agent goes left, he will avoid the hole with p=100%, then he has p=33% of going back but where he can then act just like for the first step, p=33% of not moving and then p=33% of going forward by going down.



Same thing here, the agent will go for the opposite direction of the hole, he either doesn't move, goes back, and repeat the previous step or go right and continue.



The strategy here is the same, avoid the hole with p=100%, go back with p=33% or move forward with p=66%.

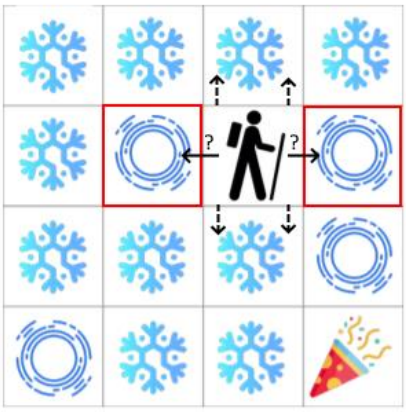


The next states are similar, the agent is in a small "loop" where the holes can be avoided by applying the same strategy as before. The only problem could come for the grey state where the agent has p=33% of going up and being in one the worst state. Let us follow the case where the grey agent has moved up.

This case will produce two dominant strategy and one occasional:

```
[['Left' 'Up' 'Up' 'Up']
 ['Left' 'Hole' 'Left' 'Hole']
 ['Up' 'Down' 'Left' 'Hole']
 ['Hole' 'Right' 'Up' 'End']]
```

```
[['Left' 'Up' 'Up' 'Up']
 ['Left' 'Hole' 'Right' 'Hole']
 ['Up' 'Down' 'Left' 'Hole']
 ['Hole' 'Right' 'Down' 'End']]
```

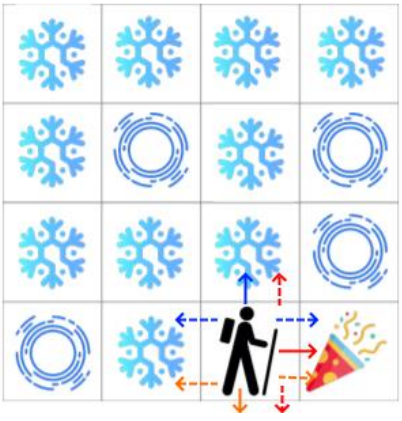


The chosen action result in a gamble with p=33% that the agent will fall in a hole, p=33% to go back to the grey state and p=33% to go back near the start. These are the dominants variants, although the agent may develop a strategy where the chosen action doesn't make him go back near the start so he will just go down:

```
[['Left' 'Up' 'Up' 'Up']
 ['Left' 'Hole' 'Down' 'Hole']
 ['Up' 'Down' 'Left' 'Hole']
 ['Hole' 'Right' 'Down' 'End']]
```

This action will gamble with p=33% that the agent will go back to the grey state and with p=66% to fall into a hole and lose. Note that this strategy is an occasional variant that appears when the agent is sufficiently lucky to not go through this state at all and to succeed with his few explorations to go back down.

As for the "last state" before the reward, where the agent is in front of reward. They are one dominant action and two possible actions.



```
[['Left' 'Up' 'Up' 'Up']
 ['Left' 'Hole' 'Right' 'Hole']
 ['Up' 'Down' 'Left' 'Hole']
 ['Hole' 'Right' 'Down' 'End']]
```

(dominant/orange)

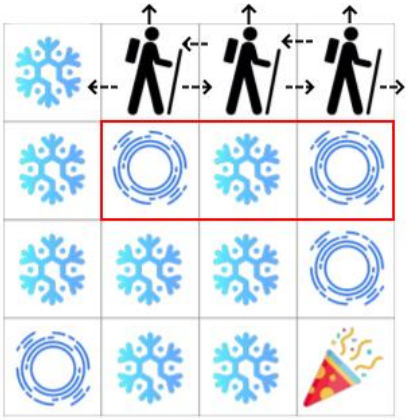
```
[['Left' 'Up' 'Up' 'Up']
 ['Left' 'Hole' 'Left' 'Hole']
 ['Up' 'Down' 'Down' 'Hole']
 ['Hole' 'Right' 'Up' 'End']]
```

```
[['Left' 'Up' 'Up' 'Up']
 ['Left' 'Hole' 'Left' 'Hole']
 ['Up' 'Down' 'Left' 'Hole']
 ['Hole' 'Right' 'Right' 'End']]
```

(Possible but not likely (~1/20) blue and red)

For this state the orange action is dominant (chosen more times) because it avoids the difficult grey situation completely, as it is one of the worst states for this strategy. Blue and red are equivalent with red slightly being more likely (still far from orange) than blue as red as p=33% to not move while blue will always move meaning if the agent misses, he will need to do more actions.

Finally, they are three cases where the previous strategy is applied, the agent will try to go back to the start and not go through the chokehold:



Note that sometimes this strategy can occur or differs:

```
[['Left' 'Up' 'Left' 'Down']
 ['Left' 'Hole' 'Right' 'Hole']
 ['Up' 'Down' 'Left' 'Hole']
 ['Hole' 'Right' 'Down' 'End']]
```

Because this path is as much as possible avoided by the IQL learner, they didn't learn a great action, going down into the hole. During training the chosen action may have been down but the agent got saved by the stochastic environment and never got the chance to change his belief. Therefore, a potent strategy may never be developed for these states such as:

```
[['Left' 'Up' 'Left' 'Up']
 ['Left' 'Hole' 'Left' 'Hole']
 ['Up' 'Down' 'Left' 'Hole']
 ['Hole' 'Right' 'Down' 'End']]
```

or

```
[['Left' 'Up' 'Left' 'Left']
 ['Left' 'Hole' 'Left' 'Hole']
 ['Up' 'Down' 'Left' 'Hole']
 ['Hole' 'Right' 'Down' 'End']]
```

Like it was previously said, most variations come from the unwanted path, there isn't a well-defined strategy here and the agents may not always learn the most efficient way of navigating this section of the environment.

Multi-agent RL: stochastic hill-climbing game

For this part two independent Q-learning agents will be implemented, they will try to coordinate in a stochastic hill-climbing game. Two commitment learners will then be implemented and both type of learning progress and how they managed to coordinate to the optimal joint action will be analyzed to see how they fare against the stochastic hill-climbing game.

Given our matrix games environment, the given payoff matrix is:

	b_1	b_2	b_3
a_1	$(16, 22, -5)$	$(4, 6, -100)$	$(10, 20, -30)$
a_2	$(4, 6, -100)$	$(25, 0, -4)$	$(10, 5, 3)$
a_3	$(8, 12, -20)$	$(10, 20, -30)$	$(4, 5, 6)$

Table 2: Stochastic Climbing Game 2: (x, y, z) means a $1/3$ chance of getting either x, y or z as a payoff.

x,y and z are equiprobable so they can be averaged as an average payoff matrix (furthermore, let us change the action number as it will be easier to implement i.e., read with the array and list starting at the index 0):

	b0	b1	b2
a0	11	-30	0
a1	-30	7	6
a2	0	0	5

On average, the joint action a0,b0 gives the most payoff, but this joint action is surrounded by two of the worst average payoffs, this will imply that with no coordination, joining forces on this action will be hard.

Each type of learners will have 1000 steps to converge to a strategy and the training will be repeated 100 times to average the probabilities to reach the joints action.

Q-learning agents:

Parameters:

- Epsilon max = 0.5
- Epsilon min = 0.
- Decay rate = 0.6

The agents, to be able to construct a viable strategy, will not explore too much, because the environment is stochastic and the agents do not coordinate, they could form a wrong belief on a certain action and miss the best joint actions by their excessive exploration.

Over 100 runs of 1000 steps, with the above parameters the independent Q-learning agents (IQL) managed to coordinate to the optimal joint actions with a probability of ~0.3%.

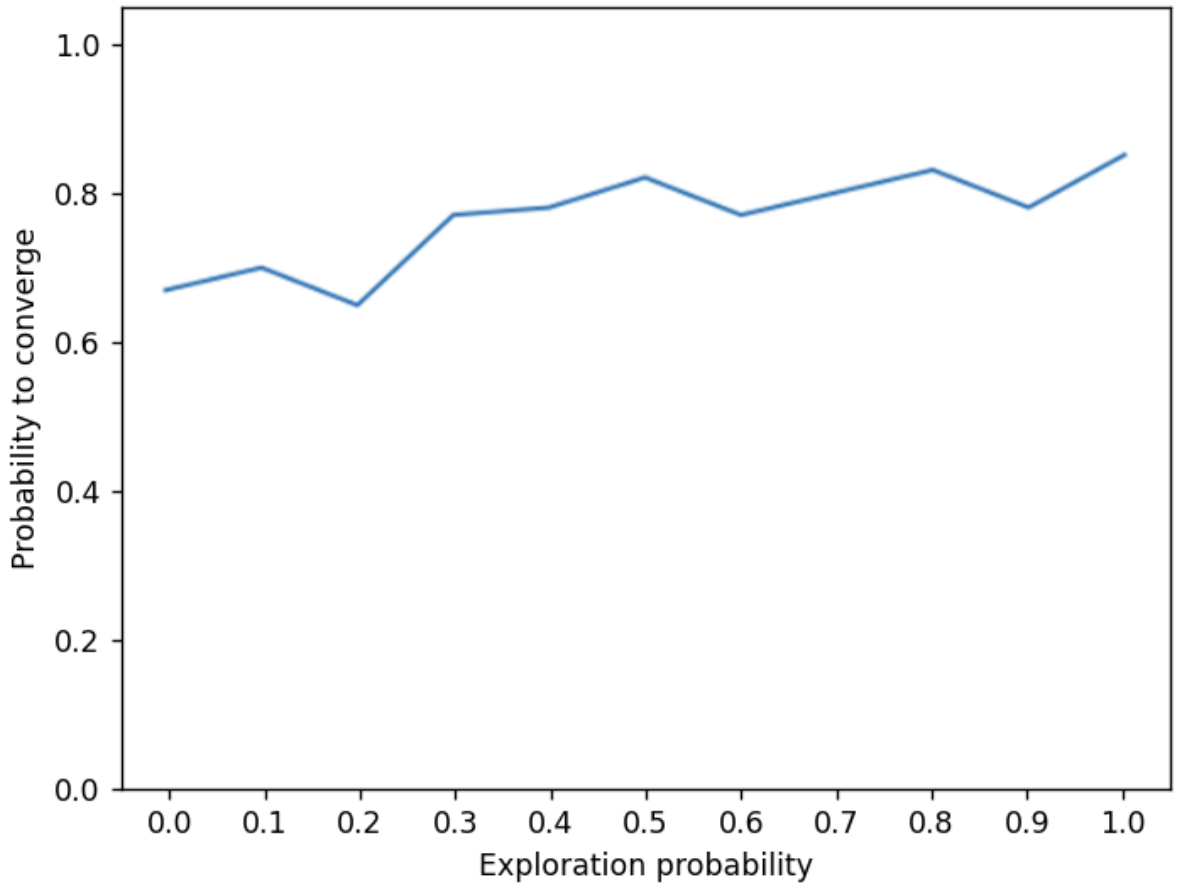
This can be explained because of the randomness of the action, player a may choose the best action to create the optimal joint action but if player b decides to explore something else than action 0, action 1 for example this will result in a punishment for both of them leading to player a avoiding action 0 and if player b explore action 0 because player a learned that from his perspective it is not great if he acts greedily and does action 1 they will get punished, pushing again player a away from an action (0 and 1) and pushing player b away from action 0. This will result in them avoiding the action leading to an optimal joint action.

This poor result is due to the exploration as it both a blessing and a curse, if by chance they coordinate in exploring their respective action leading to the optimal joint action it does not mean that they will act greedily next and nail down on this action, one could explore while the other one act greedily changing their belief about the action. Furthermore, with the epsilon decay, when it will come the time to not explore, they will not have formed a good enough belief on what actions is best. It will come down to luck of performing the optimal joint action and luck in keeping it pristine.

Hence considering their natures of IQL, it is an average result, but this is not satisfactory. Therefore, commitment learner will be implemented to compare, they, in theory, will perform better because they are committed to a sequence and the randomness will only come from the stochastic environment and not the agents themselves.

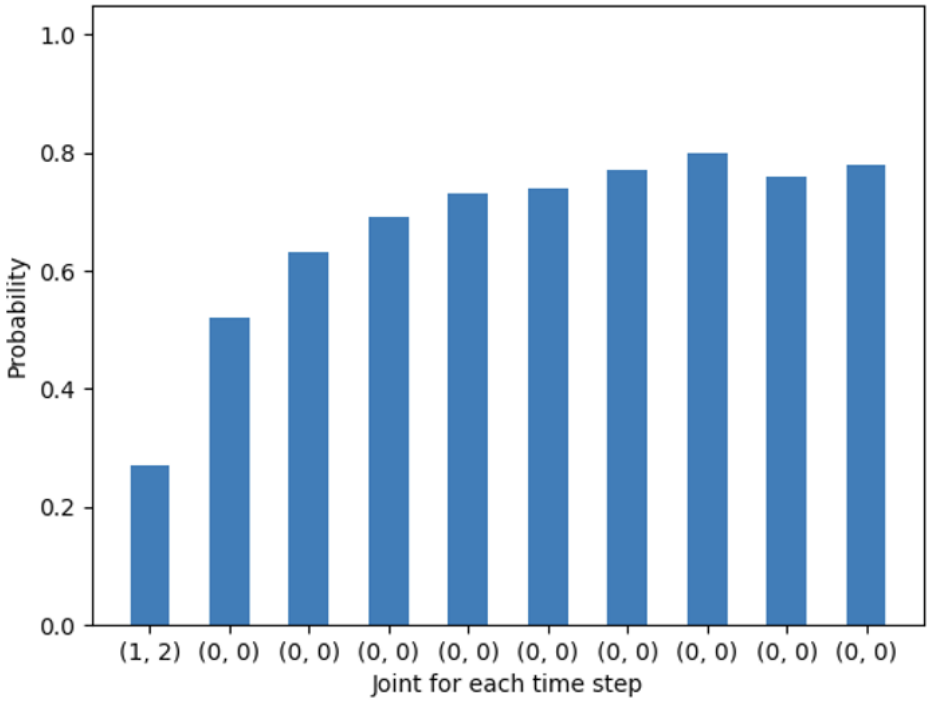
Commitment learner agents:

Like it was previously said, the randomness will come from the environment and not the agents because they will stick to a certain action on certain t frame.



The higher the probability that the agent explores, the more likely it is to find the optimal joint action. As p grows, it is more and more likely (when producing a new sequence) to produce a new combination that has never been done before and that may be optimal. Whereas, with a small p, it will produce new sequences with already explored combination that may not be optimal.

To better understand this phenomenon, take the averaged best considered joint action at each evaluation (1 evaluation every 100 episode) for p=0.9 which will be used later to compare it to the IQL.

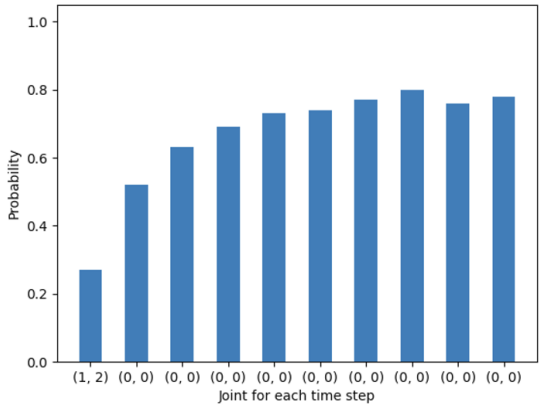


The first most frequently occurring joint action at evaluation is (1,2) with a probability to occur of 0.3, which is not the optimal joint action. Farther down, the most frequent occurring actions becomes the optimal and with an increasing probability. This proves that at each evaluation the belief of both agents that (0,0) is the best joint action is reinforced. With a p=0.9 they will explore a lot and only copy the best action of an already existing sequence with p=0.1. Meaning despite them exploring some, if not all actions, they will often try the joint action (0,0), therefore reinforcing their belief that if they need to "exploit" they will have the best joint actions.

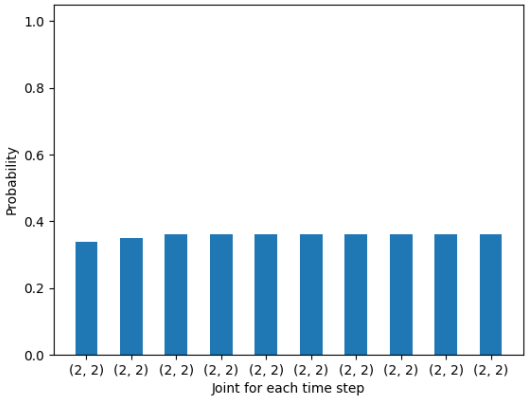
IQL vs Commitment:

Evolution through time evaluation

Commitment learner and IQL evolution will be compared through time over each of their evaluation step joint action.

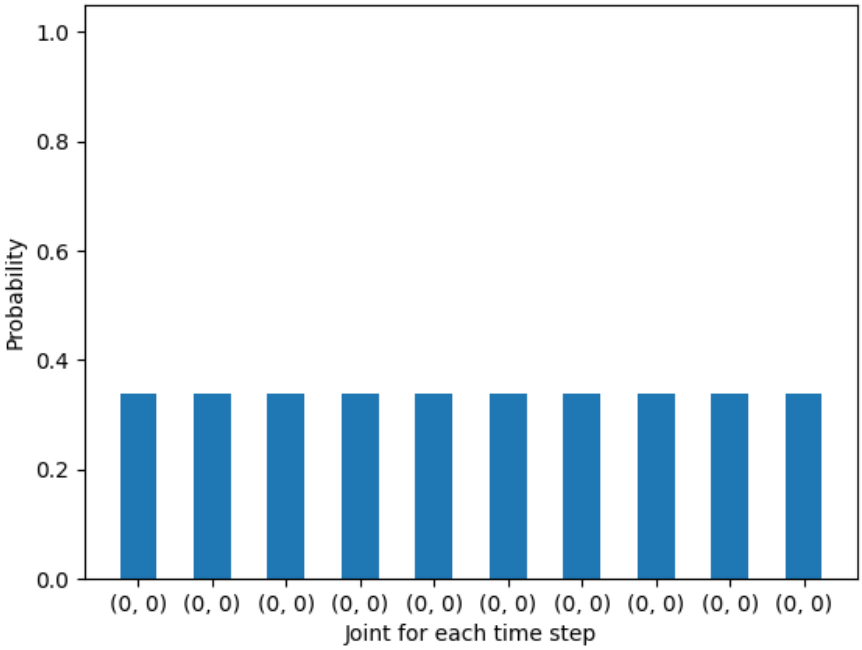


(Commitment)



(IQL)

The IQL didn't manage to have his most common "exploit" joint as the optimal joint action although sometimes he does:



But this is more due to luck than training (see the IQL section). The IQL will not coordinate and therefore will make the climbing difficult, because if one decides to do one part of the optimal joint action the other may not:

	b0	b1	b2
a0	11	-30	0
a1	-30	7	6
a2	0	0	5

If the first agent decides to do action 0 but the seconds decides to explore or just do its greedy action because they may already have been playing and he's had better rewards for action 1. They will both get an "average" punishment not knowing necessarily why. Therefore, they will converge to the joint action (2,2) as if the agent a plays action 2, he will never get punished for any action of agent b and may sometimes get rewarded.

It is the same for agent b, he will never get punished when playing action 2 for any action of agent a. He even has a reward for two of the players 1 actions, putting more emphasis on this action as the better one.

As for the commitment learners, they know that the variation in their rewards is due to the environment and therefore they can coordinate to the optimal joint action with more and more confidence.

Evolution through time training

Finally, their evolution through training will be investigated by giving the most frequent occurring joint action per 100 episodes/steps.

Note that the code has been deprecated because it serves no other purposes. Furthermore, to preserve the signature of the others functions so that the test can run smoothly, the functions signature needed to be modified, or new functions created. (Please see the annex for a look at the code)

To save the most common joint action at each time step, it needs to be calculated and saved with its probability at each evaluation, if the evaluation occurs every 100 episodes, the most common joint action will be fetched every 100 episodes. The code is nearly the same for the IQL training so it will not be added here.

For the commitment learner the following occurs (p=0.9):

```
[(0, 1), (2, 1), (2, 1), (2, 1), (2, 1), (0, 0), (0, 0), (0, 0), (0, 0), (0, 0)]
[0.32, 0.27, 0.26, 0.22, 0.2, 0.21, 0.21, 0.21, 0.22, 0.26]
```

With a more pleasant form:

0,1	2,1	2,1	2,1	2,1	0,0	0,0	0,0	0,0	0,0
0.32	0.27	0.26	0.22	0.2	0.21	0.21	0.21	0.22	0.26

For the IQL:

```
[(2, 2), (2, 2), (2, 2), (2, 2), (2, 2), (2, 2), (2, 2), (2, 2), (2, 2), (2, 2)]
[0.37, 0.76, 0.92, 0.98, 0.98, 0.99, 1.0, 1.0, 1.0, 1.0]
```

With a more pleasant form:

2,2	2,2	2,2	2,2	2,2	2,2	2,2	2,2	2,2	2,2
0.37	0.76	0.92	0.98	0.98	0.99	1.	1.	1.	1.

To construct a step-by-step visualization of the climbing, the average payoff matrix will be displayed with the commitment learner joint action in green and the IQL joint action in red and orange representing them having the same joint action.

- 0-99 steps:

	b0	b1	b2
a0	11	**-30 (0.32)**	0
a1	-30	7	6
a2	0	0	**5 (0.37)**

The IQLs chose for the first 100 steps the joint action (2,2) during 37% of their actions. Whereas the commitment learners chose (a0,b1) meaning there first commitment sequences were not great.

- 100-199 steps:

	b0	b1	b2
a0	11	-30	0
a1	-30	7	6
a2	0	**0 (0.27)**	**5 (0.76)**

The IQL's reinforced their belief that the (2,2) joint action is the best as they used it more frequently (76% of the time). The commitment learner found a new sequence (2,1) that is better and started focusing on this one.

- 200-299 steps:

	b0	b1	b2
a0	11	-30	0
a1	-30	7	6
a2	0	**0 (0.26)**	**5 (0.92)**

The IQL's didn't change and kept on going for the safe route (2,2). The commitment learners started to use a little less of their newly found joint action, as if they started a new sequence that seems better. Because do remember that they have an increasing time interval between each successive time slots, so a new sequence will have to wait longer until it is considered as a potent sequence.

- 300-399 steps:

	b0	b1	b2
a0	11	-30	0
a1	-30	7	6
a2	0	**0 (0.22)**	**5 (1.)**

The IQL's are now 100% acting greedily and think that (2,2) is the best joint action. The commitment learners are using less of their sequence with the joint action (2,1) meaning they may have started a new sequence with a different joint action, or a better sequence is getting replicated.

- 400-499 steps:

	b0	b1	b2
a0	11	-30	0
a1	-30	7	6
a2	0	**0 (0.2)**	**5 (1.)**

No change for the IQL's they will be stuck at (2,2). As for the commitment learners they are still using less and less their sequence of the joint action (2,1).

- 500-599 steps:

	b0	b1	b2
a0	**11 (0.21)**	-30	0
a1	-30	7	6
a2	0	0	**5 (1.)**

The commitment learners now do a new joint action (0,0) our optimal joint action, meaning they are "more sequences" in each agent that contains the action 0 either through replication of the most rewarding sequence that they have containing 0 as an action or through exploration.

- 600-699 steps:

	b0	b1	b2
a0	**11 (0.21)**	-30	0
a1	-30	7	6
a2	0	0	**5 (1.)**

The commitment learners are honing their knowledge by repeating the sequences, exploring possible combinations, and therefore building if (0,0) is on average more interesting, that it wasn't a fluke where they got very lucky for the times, they tried this sequence but on average it wouldn't be great.

- 700-799 steps:

	b0	b1	b2
a0	**11 (0.21)**	-30	0
a1	-30	7	6
a2	0	0	**5 (1.)**

Same thing as before.

- 800-899 steps:

	b0	b1	b2
a0	**11 (0.22)**	-30	0
a1	-30	7	6
a2	0	0	**5 (1.)**

The commitment learners are now doing more actions 0 on their own sequences resulting in more joint action (0,0) being observed, they may now have considered it to be sufficient and reliable to go climb up the ladder in this direction.

- 900-999 steps:

	b0	b1	b2
a0	**11 (0.26)**	-30	0
a1	-30	7	6
a2	0	0	**5 (1.)**

Finally, they are frequently using the actions 0 meaning they are more sequences with action 0 on both agents resulting in more and more of the joint actions (0,0).

If the agents were asked to be greedy, their answers would probably be for both actions 0 as they have had enough time steps with sequences that result in a joint action (0,0) to know that it should be the best action.

Note: the creation of sequences and their associated actions is random, meaning the above observation and interpretation of the growing probability of the joint action (0,0) for the commitment learner can be either due to the agents replicating the action of the most successful sequence with p=0.1 or trying a random action with p=0.9 and both getting 0 for this sequence. The most occurring joint action per 100 steps doesn't give much information for the commitment learner with a high exploration rate the actions are random for most of the sequence that will be seen; but with a lower exploration rate, the sequences are more likely to have been created through replication, and therefore the best considered action grows in "popularity". A great indicator for high exploration rate will be "what is seeing at each evaluation" because as the timestep increases the more likely the agents are to have tried most sequences and here only the best considered at each evaluation will be displayed.

Interpretation:

IQL:

The IQL agents have no coordination, when player a decides to explore and do action 0 and player b decides to explore and do action 1, they will get an average payoff of -30 meaning player a will learn to avoid action 0 not knowing that it is because of player's a action. Now that player a will avoid action 0, if he decides to do action 1 and player b explore and does action 0, they will get an average payoff of -30 and the same will happen, player b will learn to avoid action 0.

In the case where player a and b coordinate by chance and explore both action 0 at the same time. They will learn that this is a good action, but if player a act greedily and does action 0 but player b explores and does action 1, player a will not know that if it is due to the environment or the other player, he will think that action 0 is not that great when it was due to player b exploring and doing action 1. Therefore, they will avoid these actions.

	b0	b1	b2
a0	**11**	**-30**	**0**
a1	**-30**	7	6
a2	**0**	0	5

This will create like a wall, where because they do not know what the other player played consistently* and will climb down towards a stable action 2, where even if the other player explores, they will never get punished and therefore not avoid this action.

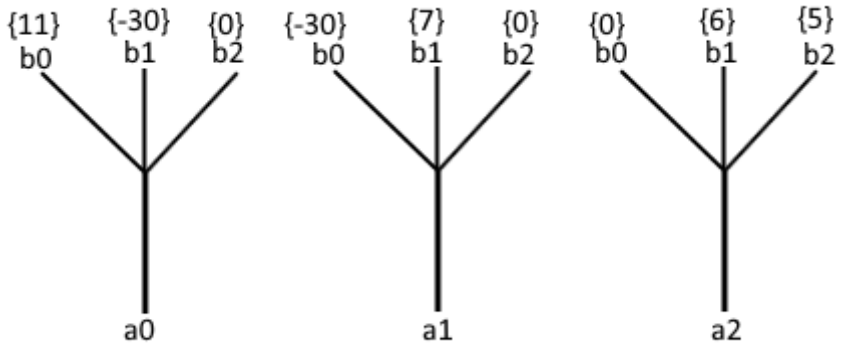
*(note that the commitment learner does not know what the other player played either, but they can "guess" because they are consistent at each sequence time step)

Commitment learner:

The commitment learners have coordination; their success rely on their ability to predict the reward by sticking to a sequence while developing new sequence with either a random action or the action of the best "working" sequence. When arriving at the timestep of a specific sequence they know that for this sequence the other did the same action and didn't derive. Therefore, each player is reliable, and this eliminates the randomness of the other players actions, only the stochasticity of the environment remains but they can prevail thanks to their commitment and learn the average payoff of this joint commitment.

Thought and conclusion:

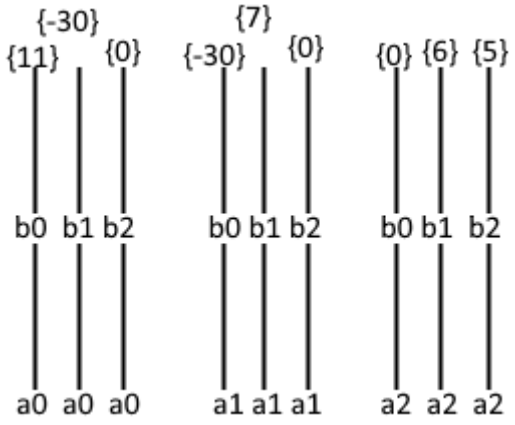
The stochastic hill-climbing game is a great example of how unfit basic IQL agents are for cooperative game. A simple analogy is a reference to the name of the game, "hill-climbing". Each agent needs the other to climb the hill, they need the other to be reliable. This analogy can be expressed as such:



Note that it is not a turn game, but this representation helps to understand from the point of view a single agent. The agent doesn't know the other player action, the rewards flow down the path and goes to the action reward.

Player a will only see the rewards he gets from his action and not the player b actions, if player b acts erratically (by exploring) and is not therefore reliable, consider that b has p=1/3 to do either action b0, b1 or b2. The erratic behavior of b leads to an average payoff of -6.33 for action a0, -7.66 for a1 and 3.66 for player b "exploration phase". Of course, this implies the erratic behavior of player a too because their exploration rate is the same, so it is the same from the perspective of b. Both players due to their random explorations at random episode (with a decreasing probability) leads to them being unable to "trust" the other and climb the hill and stay at the bottom with action 2.

Where the commitment agent shine is in eliminating the randomness of the action and therefore building trust, the path is now:



Note that it is not a turn game, but this representation helps to understand from the point of view a single agent. A line represents the chosen action for a sequence, the agent doesn't know what the other chose but that a specific reward flows down from this sequence.

Player a knows that if he picked for a certain sequence the action 0 that player b will do an unknown action but the same one for each step of the sequence meaning player a will have an overall idea of the payoff. This is the way commitment builds trust, when they will try a new sequence, they will now that if the combination of action doesn't work it is because it is not viable and not because the other player didn't want to follow "the plan". Hence, this is how they reach cooperation and manage to succeed more frequently.

In conclusion, the IQL agents are not working against each-other but not trustworthy and cannot (often) climb the hill together through cooperation whereas the commitment learners are trustworthy and can climb the hill together in this cooperation game.

Annex:

Function for the most occurring joint action per 100 episodes:

```
def train_in_depth_commitment(env: MatrixGame, t_max: int, n_min: int, n_init: int,
                              p: float, evaluate_every: int) -> Tuple[List,List]:
    agents = [CommitmentAgent(env.num_actions, t_max, n_min, n_init, p) for _ in range(env.num_agents)]
    E_joint=[]
    Joint_per_100=[]
    prob_joint_per_100=[]
    for episode in range(t_max):
        actions = np.zeros(env.num_agents,dtype=int)
        for agents_nb in range(env.num_agents):
            actions[agents_nb] = agents[agents_nb].act(episode)
        reward = env.act(actions)
        for agents_nb in range(env.num_agents):
            agents[agents_nb].learn(episode,reward)
        E_joint.append((actions[0],actions[1]))
        if (episode + 1) % evaluate_every == 0:
            Joint_analysis={}
            for action in E_joint:
                Joint_analysis[action]=Joint_analysis.get(action,0)+1
            mst_common_act=max(Joint_analysis,key=Joint_analysis.get)
            Joint_per_100.append(mst_common_act)
            prob_joint_per_100.append(Joint_analysis[mst_common_act]/evaluate_every)
            E_joint=[]

    return Joint_per_100,prob_joint_per_100
```

Acknowledgements:

Many thanks to Denis Steckelmacher, Ann Nowé and Elias Fernández for the courses they have provided and the resources that were made available which made it possible to understand the notions and produce the code and this report.

I would also like to thank Andries Rosseau and Raphael Avalos for their availability, clarifications and help they have provided which were of great help for producing the code and this report.

I would also like to give credit to this thread that has been used to get an overall idea on how to create a list of a range with incremental step to build the sequences:

https://stackoverflow.com/questions/40706034/how-to-create-a-list-of-a-range-with-incremental-step