Analysis of Markov Decision Processes, Policy and Value Iterations, and Q-Learning

Chua Qi-Yang

November 27, 2021

1 Preface

In this analysis, the aim is to study the behaviors of Policy Iteration(PI) and Value Iteration(VI), and how Q-Learning as a Reinforcement Learning Algorithm can affect the strategy used to tackle the Markov Decision Processes(MDP). The MDP, PI, VI and Q-Learning are implemented using the OpenAI Gym Library. [2]

2 Markov Decision Processes

The two interesting MDP chosen were Frozen Lake for large state spaces and Quiz Game Show for the non Grid World example.

2.1 Frozen Lake

The Frozen Lake is a 20x20 grid containing 4 possible areas, Safe (S), Frozen (F), Hole (H) and Goal (G). The agent moves around the grid until it reaches the goal or a hole. As the floor is slippery, there is a chance that the agent moves in a perpendicular direction that was not intended. If it falls into a hole, it has to start from the beginning and is rewarded the value -1. The process continues until it learns from every mistake and reaches the goal eventually.

2.2 Quiz Game Show

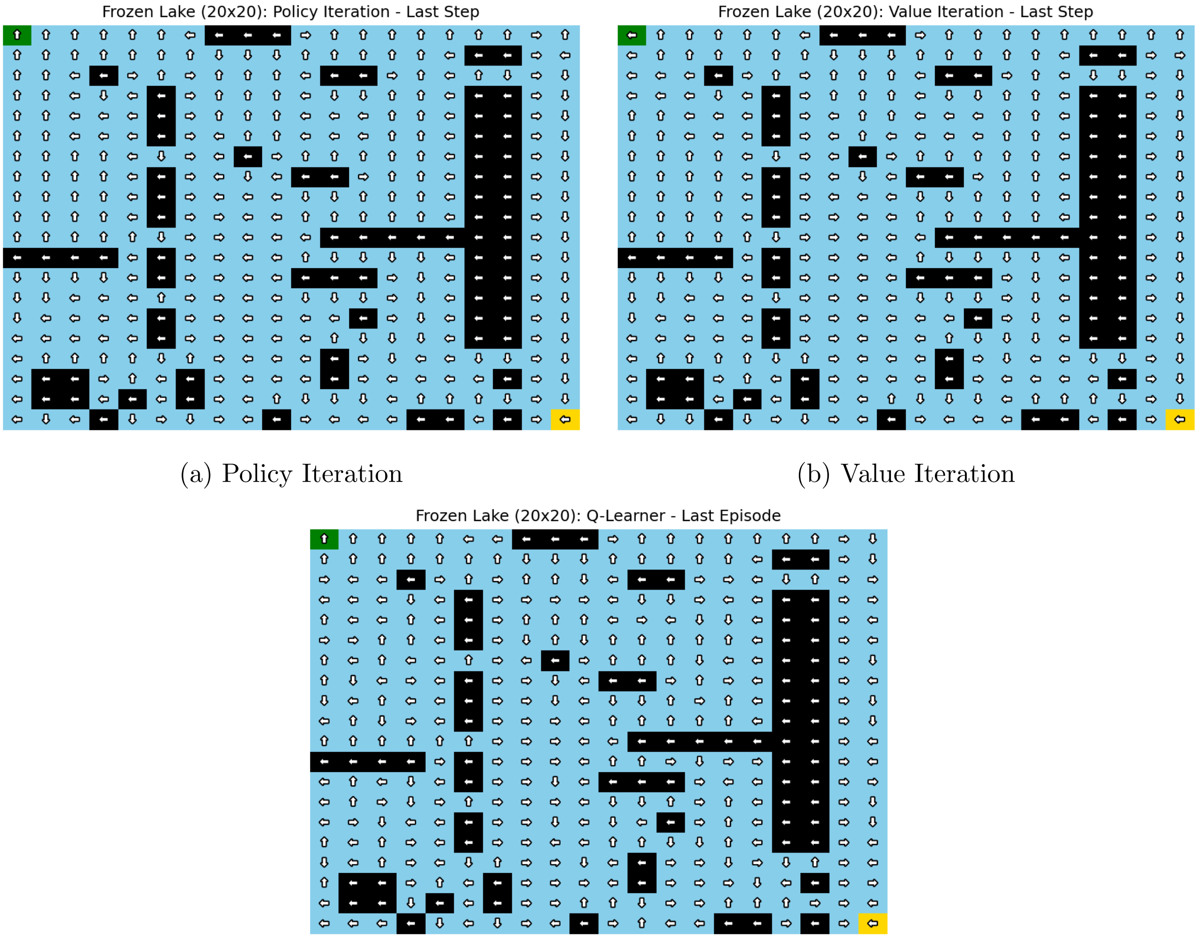
Quiz Game Show is an interesting MDP that was found while searching for interesting MDP not set in a Grid World. [1]

In a quiz game show there are 10 levels. At each level, one question is asked and if answered correctly, a certain monetary reward based on the current level is given. Higher the level, tougher the question but higher the reward.

At each round of play, if the participant answers the quiz correctly then s/he wins the reward and also gets to decide whether to play at the next level or quit. If they choose to quit, then the participant gets to keep all the rewards earned so far. At any round if participants failed to answer correctly then s/he looses "all" the rewards earned so far. The game stops at level 10. The goal is to decide on the actions to play or quit maximizing total rewards.

3 Policy and Value Iterations and Q-Learning

3.1 Frozen Lake


Frozen Lake (20x20): Policy Iteration - Last Step

Frozen Lake (20x20): Value Iteration - Last Step

(a) Policy Iteration

(b) Value Iteration

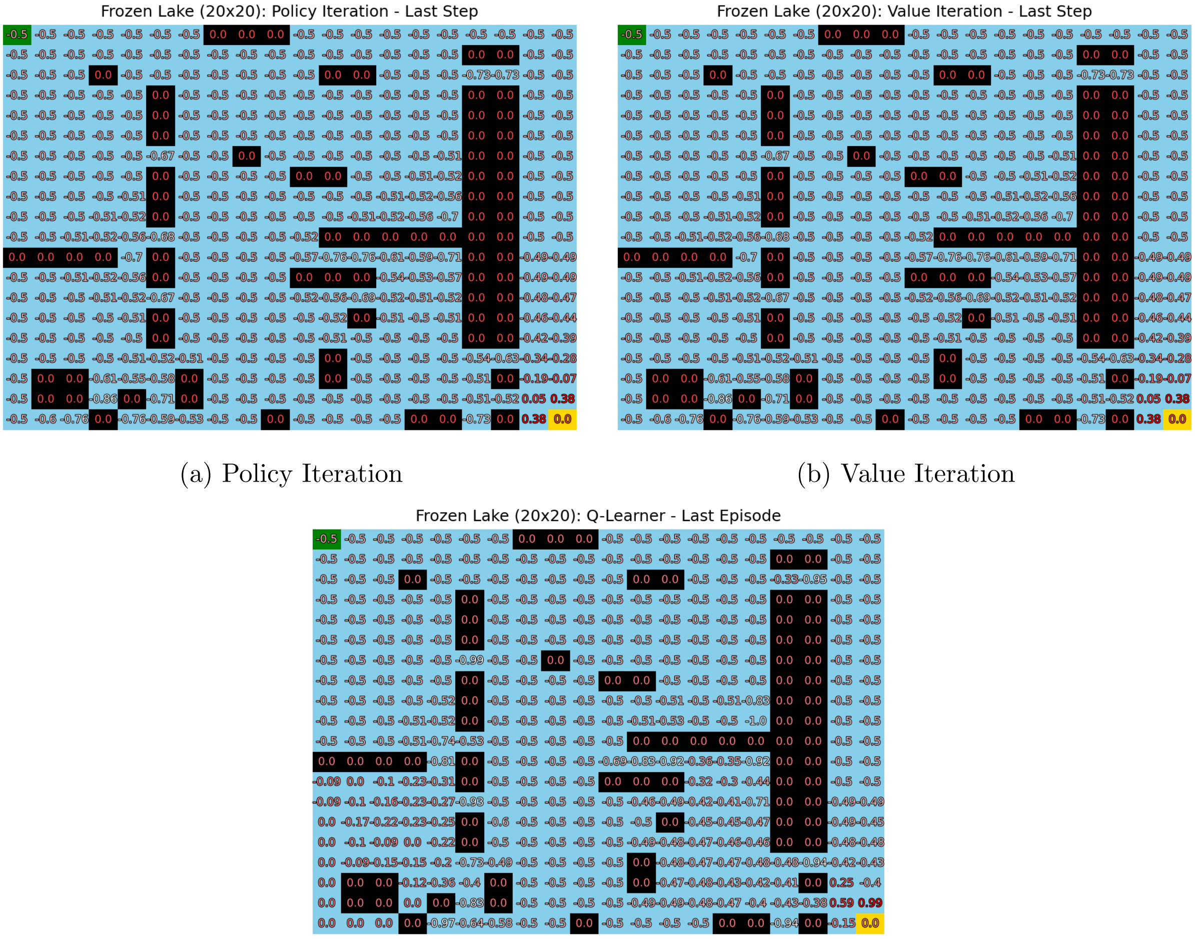
Frozen Lake (20x20): Q-Learner - Last Episode

(c) Q-Learning

Figure 1: Final Policy

For the Frozen Lake MDP, it seems that both the PI and VI eventually opted for similar general policies, opting to stick to the sides of the grids and depending mostly on the slip factor of the Frozen Lake problem to move towards the goal. This seems to be making use of the fact that sticking next to the walls will also reduce the number of directions that the agent can slip towards, which inherently reduces risk of moving into a grid that is unfavourable. This behavior is also partly induced by the fact that the slip factor in OpenAI Gym's Frozen Lake example causes the agent to slip more frequently than going in the intended direction. The chance of going in the intended direction is the same as going in either of the perpendicular directions, making each possible moves $\frac{1}{3}$ probability.

As Q-Learning did not manage to converge, there are some policy decisions made by Q-Learning that



(a) Policy Iteration

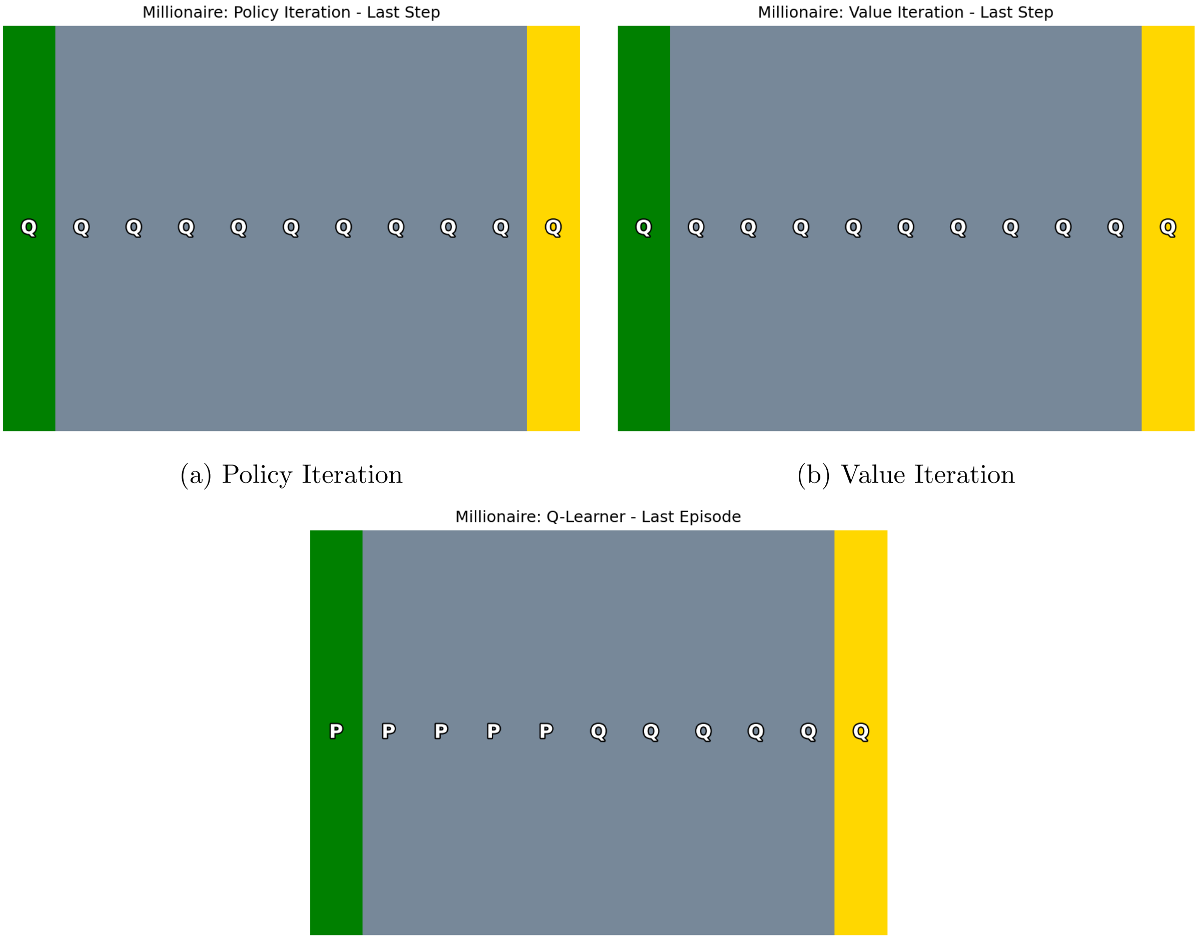
(b) Value Iteration

(c) Q-Learning

Figure 2: Final Value

are quite interesting. For example, on the left side of the grid on row 13, there is an arrow that is pushing the agent towards a hole. Initially, I figured that due to the lack of convergeance, this might be a mistake made by the algorithm. However, it might be possible that the algorithm decided that the area it was in is hard to navigate out of successfully without dropping into the holes. So it took advantage of the fact that when you drop into a hole, it will return you to the start, which might be a more favourable position to the algorithm, despite the penalty of -1. Another interesting behavior is the grid above the goal, which made the agent move away from the goal instead of going into it. Looking at the values generated by the algorithm, it favoured the grid very significantly compared to any other grid. This might be a reason why it would prefer to move between two of the highest valued grids and not enter the terminal grid.

In terms of computation, PI took 7 steps to converge, while VI took 35 steps. Q-Learning did not manage to converge even after 1000 steps. This might have been caused by the huge random factor caused by the slip factor.


Millionaire: Policy Iteration - Last Step
Q Q Q Q Q Q Q Q Q Q Q
Millionaire: Value Iteration - Last Step
Q Q Q Q Q Q Q Q Q Q Q
Millionaire: Q-Learner - Last Episode
P P P P P Q Q Q Q Q Q

(a) Policy Iteration

(b) Value Iteration

(c) Q-Learning

Figure 3: Final Policy

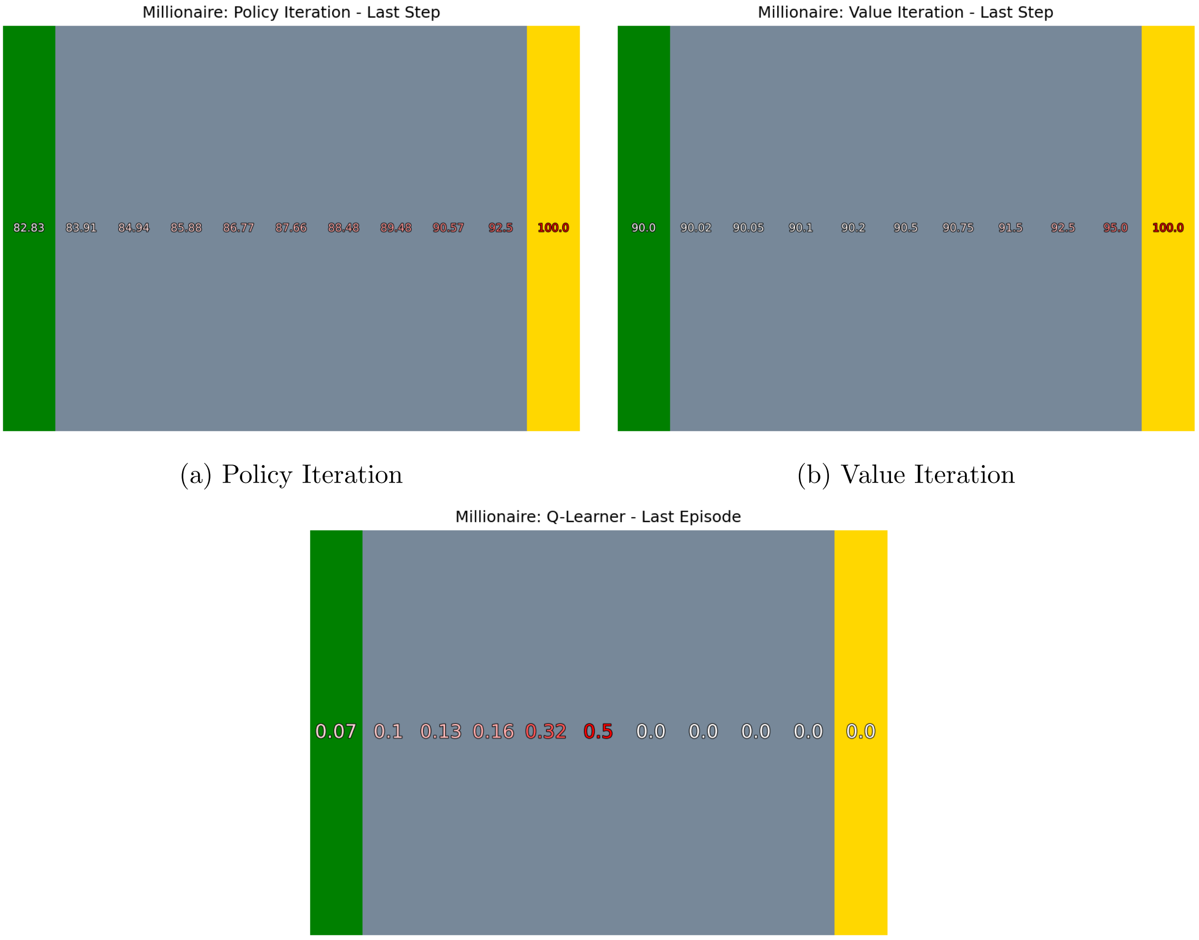
3.2 Quiz Game Show

Quiz Game Show produced very surprising results. Both Policy and Value Iterations decided that the best plan was to quit the game straight away, even though the first question had a 99% chance of success. As the Quiz Game Show environment was coded by myself with references to the Frozen Lake example from OpenAI Gym, it might have caused some issues with the way the actions are weighted against each other. This can be seen from the extremely high values assigned to each of the states, which diminished the incentive to take the risk when the first state was already weighted at ¿80, with the last state weighted at 100.

Q-Learning provided a more realistic approach to the game, opting to play for 5 rounds before the policy was to quit at that point.

Further improvements to the Quiz Game Show environment should be possible to help bring the Policy and Value Iterations to a more logical approach to the game instead of just quitting from the start.

In terms of computation, PI took 1 step to converge, while VI took 133 steps. Q-Learning still did not manage to converge even though Quiz Game Show was a less complex environment compared to the Frozen Lake.



(a) Policy Iteration

(b) Value Iteration

(c) Q-Learning

Figure 4: Final Value

3.3 Performance

Below are the computation times taken by each of the methods over both of the MDP:

- PI: 284 seconds
- VI: 35 seconds
- Q-Learning: 3216 seconds

VI performed the fastest, taking almost $\frac{1}{100}$ of the time taken by Q-Learning. However, this is to be expected as Q-Learning has much more complex calculations to be done for each case compared to PI and VI.

PI took a relatively longer time than VI, which is to be expected given that VI makes use of random values to quickly calculate them based on neighbors and quickly achieve convergeance. Whereas PI depends on mapping states to actions, which can be time consuming when dealing which large state spaces such as the Frozen Lake example.

From the above examples, it seems that even though Q-Learning can take much longer than PI and VI, as a Reinforcement Learning Algorithm, it is still able to come up with strategies that diverge from PI and VI, while at the same time making sense when assessed by humans.

References

[1] Somnanth Banerjee. Real world applications of markov decision process. https://towardsdatascience.com/real-world-applications-of-markov-decision-process-mdp-a39685546026. (Accessed: 26.10.2021).

[2] OpenAI. Gym library. https://gym.openai.com/. (Accessed: 26.10.2021).