ML Assignment 4

Kuhu Gupta

1 Introduction

This assignment task investigates reinforcement learning. It starts by picking two Markov Decision Processes (MDPs) and solved utilizing policy iteration and value iteration with outcomes looked at against one another. Thereafter, Q-learning is connected to the two chosen MDPs and those outcomes are then contrasted with the first outcomes. From the discussion in the class, Markov Decision Processes is a model based on the Markovian property which implies that the following state in the wake of making a move from the present state is just identified with the present state and has no connection to past states before the present states. In my comprehension, the two most vital aspects of MDPs are its states and its transition model. The task is to locate the ideal approach to guarantee to get the best reward by taking an activity from the present state. Be that as it may, MDPs isn't constantly founded on this real world however it can give a system to create a structure for decision making when concentrating a wide scope of optimization and reinforcement learning problems. As the outcome, there are numerous applications dependent on MDPs.

2 Description of the problems

Gridworld path finding was picked as the reason to show for the two Markov Decision Processes. This was accomplished for a couple of various reasons. To start with, the simplicity in visual portrayal and comprehension. Second, path finding is a typical issue, in the real world that has applications from strolling to work, to driving cross-country, and so on. Third, it is not challenging to scale the unpredictability of a grid such as an increment in the quantity of states. Fourth, for a solveable matrix, at least one ideal arrangements must exist. All things considered, grid world is extraordinary for exhibiting reinforcement learning.

- The first grid world issue is a simple configuration. It has a 10x10 shape with 70 conceivable states.
- The second grid world issue is a hard configuration. It has a 20x20 shape with 353 conceivable states.

Each grid would have its begin state is in the base left corner and its end state is in the upper right corner. Different stops exist that constrain the answer for enhancing around. Different states have negative rewards associated with them and are shading covered in the state diagrams underneath.

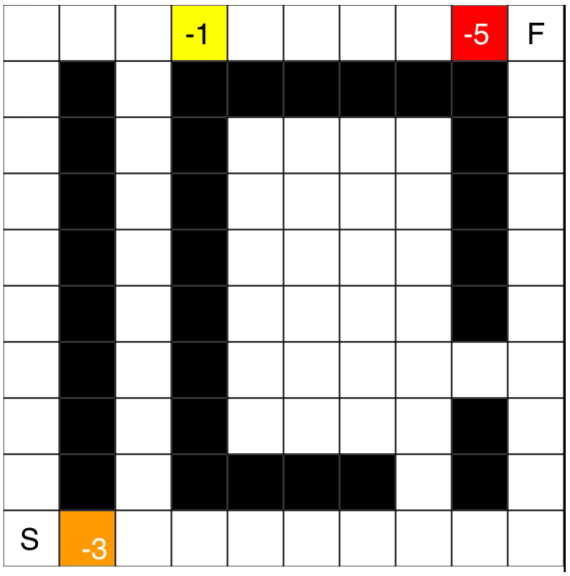

Figure 1: 10x10 Grid World with 70 States

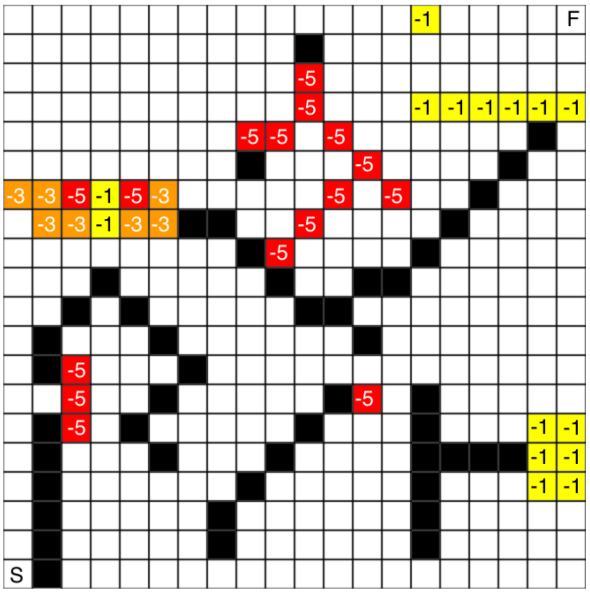

Figure 2: 20x20 Hard Grid World with 353 States

3 Overview

These are two essential strategies for unraveling MDPs. Both value iteration and policy iteration accepts that the agent realizes the MDP model of the world by knowing the state transition model and reward probability function. Along these lines, they can be utilized by the agent its further actions based on the prior information about the environment before interaction. On the contrast, Q-realizing is model-free learning condition that can be utilized in a circumstance where the operator at first realizes just that are the conceivable states and activities however doesn't know the state-transition and reward likelihood functions. In Q-learning, the agent improves its nature through gaining from the historical backdrop of associations with the environment.

3.1 Value Iteration

Firstly, value iteration is used to ideally solve the Markov Decision Process. This reinforcement learning algorithm works by recursively calculating the value function in order to get an optimal value function at the end. It utilizes the Bellman condition for traversing from one state to another state to reach convergence. Since the utility isn't at first known, the calculation starts by initializing value function with arbitrary value and with the reward at the final state and works in reverse computing utility for closest states. This proceeds until all states are assessed. After various cycles, the algorithm converge to an optimal value.

3.2 Policy Iteration

Secondly, to ideally comprehend the MDPs the second reinforcement algorithm is Policy Iteration. Policy iteration will re-characterize the policy at each progression and register the reward as indicated by this new approach until the policy converges. Policy Iteration additionally ensured to converge to the optimal policy and it regularly takes less cycles to converge than value iteration algorithm.
It works by starting with an arbitrary policy and after that take multiple attempts to alter it state by state by taking distinctive actions. This proceeds all through each state of the policy trying to find an optimal policy. At the point when no further policy changes are discovered, convergence is achieved.

4 Results of Policy Iteraion and Value Iteration

4.1 Value Iteration Results - Easy Grid World

The first MDP analyzed is the simple gridworld pathfinding test. A 10x10 framework with different obstacles. This grid world problem is a generally simple issue with a moderately modest number of complete states. For measuring convergence, a cycle limit of 1e-6 is considered. For this simple grid world problem, it takes 68 cycles for value iteration's policy delta to converge underneath the edge. This does makes sense, as the value iteration is designed to work in reverse and endeavors to figure out the most ideal approach. Upgrades in the rewards are made additional over the time and in the end for easy grid world it ends at 68 iterations.
If you look at the execution, the optimal policy is achieved at 68th iterations by getting the highest reward. On comparing 5th iteration with 40th, 60th and 68th iteration there is a suprising phenomeno to notice in the number of steps as they are almost same. This clearly hints at the fact the briefest way isn't really the most profitable. "Imagine a robot that is trying to solve a maze and there are two paths to the goal state one of them is longer but gives higher reward while there is a shorter path with smaller reward." (https://medium.com/@m.alzantot/deep-reinforcement-learning-demysitifed-episode-2-policy-iteration-value-iteration-and-q-978f9e89ddaa) On analyzing the maps,it seems that the calculation gets along admirably at deciding an ideal approach for the different conceivable states.

No.of Iterations	Used Time	Reward Gained	Steps Taken	Convergence Value
1	0.0127	-199.8600	196.370	72.86500
5	0.063	-19.5300	42.7200	21.2822
10	0.1110	8.6700	22.4000	13.3210
20	0.268	42.9800	28.94000	1.7836
40	0.3439	44.9200	42.5200	0.0010
60	0.4001	50.5600	49.1000	4.86e-6
68	0.4120	52.3200	48.3600	9.24e-7

Table 1: Easy Grid World Value Iteration Results

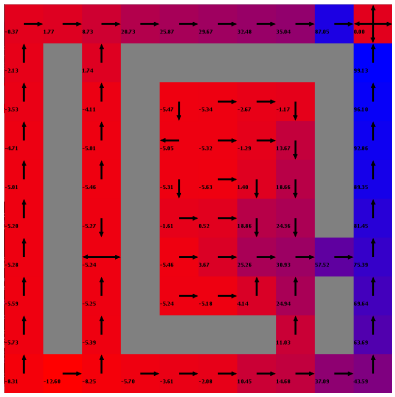

Figure 3: Value Iteration for Easy Grid World - 5 Iterations

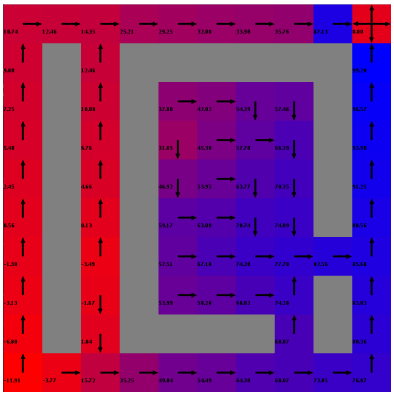

Figure 4: Value Iteration for Easy Grid World - 10 Iterations

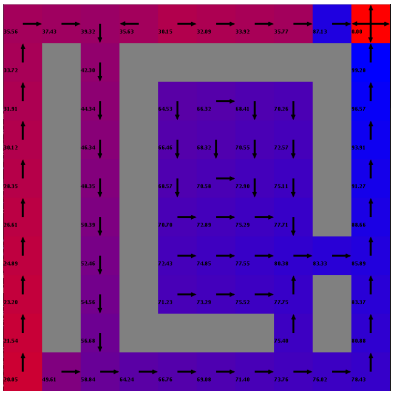

Figure 5: Value Iteration for Easy Grid World - 68 Iterations

4.2 Value Iteration Results - Hard Grid World

In this section, you will get the analysis of value iteration algorithm results for Hard grid world pathfinding test. This grid world is comparatively more intricate than the one used above due to greater number of states, rewards, and obstacles. It is a 20*20 matrix.
For this grid world also I have used the same measure of convergence value of 1e-6. As compared to the simple grid world, this complex grid world takes 64 iterations for value iteration's policy delta to meet underneath the edge. Therefore, If you compare the number of steps of easy which as 68 iterations and complex grid world which has 64 iterations, it hints at intricacy of the grid world is not responsible for the increase or decrease in the number of iterations for convergence. Moreover, in terms of duration, the complex grid world problem takes signficiantly longer to converge than the simple grid world problem. As seen in Table 2, the reward gained at 40th and 64th iteration have a very small difference with greater reward at 40th. From the maps, it turns out to be certain that the algorithm completes a somewhat magnificent activity at deciding an approach with the best activity to take at the different states.

No.of Iterations	Used Time	Reward Gained	Steps Taken	Convergence Value
1	0.0621	-305.1100	302	84.1168
5	0.187	-287	300	25.6000
10	0.3233	-286	303	13.1212
20	0.6753	-273.6300	272.4000	3.7890
40	1.1501	14.8600	63.4000	0.0076
60	1.3635	12.3000	61.3400	2.54e-5
64	1.5012	14.3900	60.4300	1.22e-6

Table 2: Hard Grid World Value Iteration Results

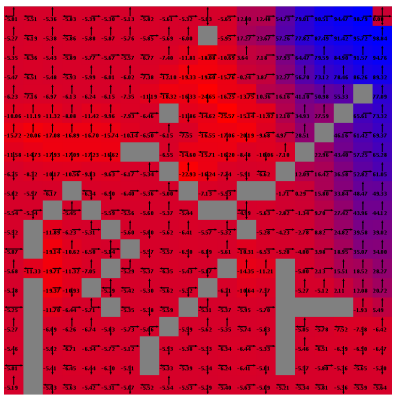

Figure 6: Value Iteration for Hard Grid World - 5 Iterations

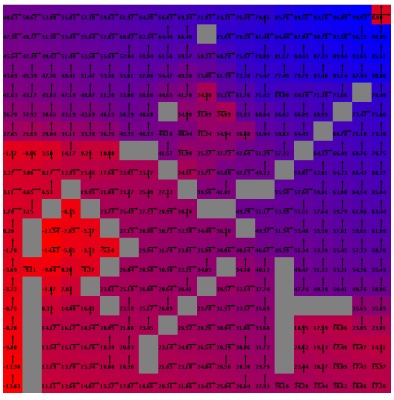

Figure 7: Value Iteration for Hard Grid World - 30 Iterations

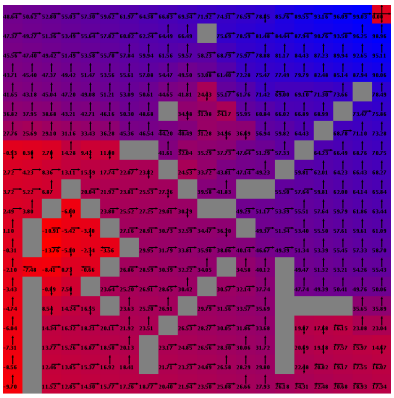

Figure 8: Value Iteration for Easy World - 64 Iterations

4.3 Policy Iteration Results - Easy Grid World

In this section, we are examining the easy grid world path finding test using the policy iteration. Similar to the value iteration for quantifying convergence, a threshold of 1e-6 is utilized. In comparison to value iteration, policy iteration take much less iteration, that is 36 for convergence which clearly states that policy iteration takes much more efficient but it does take more time per iteration. The policy iteration approach begins to converge in all respects rapidly, it begins to levels as there is no and does not see significant change seen after 10 iterations.
On looking at table 3,the greatest reward is found at 36th iteration which gives an optimal policy after convergence. In 10th and 30th iterations it seeems, the algorithms gets stuck by taking intricate steps. However, if you consider the time of the iteration, the algorithm is fairly fast which is helped by the way that just 36 stages are required for convergence. Taking a gander at the strategy maps, it is clear that policy iteration performs great for this issue.

No.of Iterations	Used Time	Reward Gained	Steps Taken	Convergence Value
1	0.0186	-185.9000	189.8560	77.3400
5	0.1154	42.9860	24.2560	24.3222
10	0.2145	46.6500	48.8900	4.0862
30	0.3002	46.5791	47.4600	1.89e-6
36	0.3041	48.0190	45.0890	9.34e-7

Table 3: Easy Grid World Policy Iteration Results

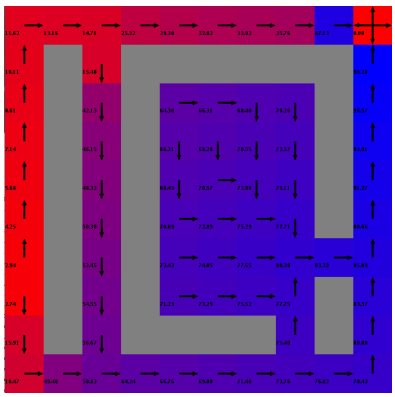

Figure 9: Policy Iteration for Easy Grid World - 5 Iterations

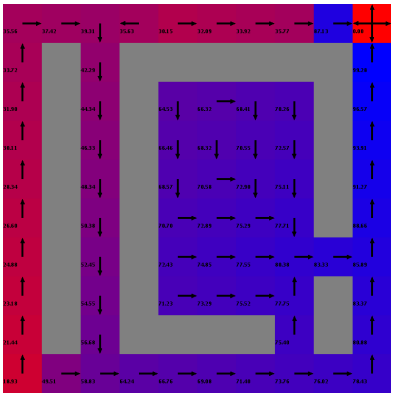

Figure 10: Policy Iteration for Easy Grid World - 10 Iterations

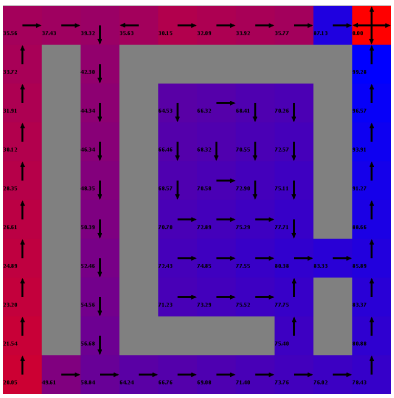

Figure 11: Policy Iteration for Easy World - 36 Iterations

4.4 Policy Iteration Results - Hard Grid World

On the contrary to the easy grid world, the hard grid world gives a much better performance by taking just 25 iterations. If you look at the convergence value, there is a sudden change encountered at 10th and 20th iteration but gradually it becomes more stable. In this arrangement where there are so many obstacles, the policy iteration approach does great as it splits itself to traverse as many states as possible which seems to be evident in the maps. Surprisingly, the highest reward is at the 10th iteration. In terms of time it only takes 2 seconds and 62 steps which kind of tells to be an optimal policy. Therefore, Policy iteration clearly outshines the value iteration.

No.of Iterations	Used Time	Reward Gained	Steps Taken	Convergence Value
1	0.4011	-296.5800	303	84.1124
5	0.9140	-247.3200	264.0234	59.9934
10	1.8654	10.5673	62.4530	2.2218
20	2.6412	9.8760	65.1230	6.43e-6
25	2.6623	9.5477	65.4340	9.03e-7

Table 4: Hard Grid World Policy Iteration Results

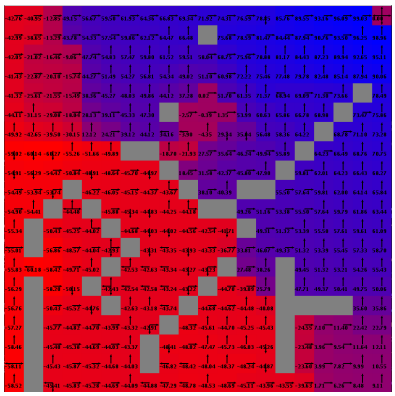

Figure 12: Policy Iteration for Hard Grid World - 5 Iterations

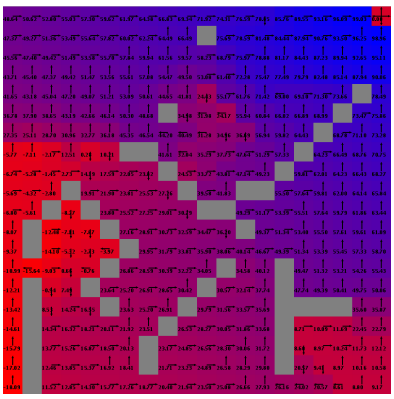

Figure 13: Policy Iteration for Hard Grid World - 10 Iterations

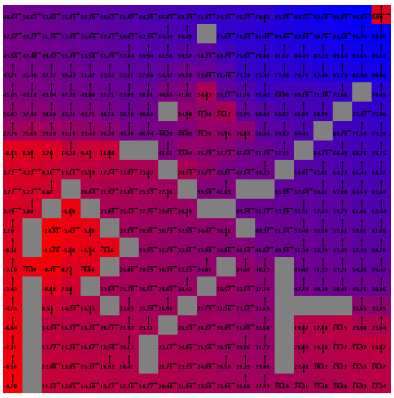

Figure 14: Policy Iteration for Easy World - 25 Iterations

5 Q-learning

5.1 Overview

Thirdly, Q-learning is used as the reinforcement algorithm for ideally solving the MDPs.Q-Learning is a case of model-free learning algorithm. It doesn't accept that agent knows anything about the state-transition and reward models. Be that as it may, the specialist will find what are the good and bad actions by experimentation. The fundamental thought of Q-Learning is to surmise the state-actions tuples Q-work from the examples of Q(s, a) that we see during interaction with the environment. In this algorithm, each state gets a q value. At each state, the student ascertains new q value dependent on both current and future prizes. At first, Q-learning will invest time in exploration, though it will in the long run streamline dependent on information learned.

5.2 Results

The outcomes acquired utilizing Q-learning on both easy and hard Grid World problems were very intriguing. On comparison to policy and value iteration, the Q-learning have no idea about the model and learns as it investigates. For each problem, a combination of parameters are utilized to tune an ideal learner. These parameters consists of learning rate, the intial q value to be set, and the epsilon parameter. For every preliminary, a most extreme number of steps to investigate for each learning scene is defined as 300. A rundown of the outcomes from shifting these parameters is shown in the table. On analyzing the table, a learning rate of 0.1, starting q value of 0, and an epsilon estimation of 0.1 stands out as the ideal parameters for both the simple and hard grid world problems.

The low q values empowered investigation as they showed unbiased introductory data. The low learning rate likewise put an accentuation on more established data which is useful in exploring the grid world successfully. The quantity of iterations and time, nonetheless, shifted significantly. For the simple gridworld issue, the ideal outcomes took 160 iterations to accomplish a greatest reward just as meet a to some degree low convergence factor. These outcomes were fundamentally the same as the ones saw from the policy approach and value iteration. Strangely, an opportunity to run the algorithm was significantly quicker. This can be ascribed to the q-learning algorithm not squandering its time investigating awful useless ways as it advances. A lower learning rate likewise ensures it uses data about important path.

For the hard gridworld issue, the ideal outcome took an any longer 2800 iteration. As the learner needed to investigate furthermore, explore a genuinely intricate grid world, this bodes well. From the approach maps underneath, obviously the calculation builds up an ideal way that it endeavors to pursue excepting haphazardly picking a misguided course. Once more, its run time was very low thought about to different methodologies for illuminating the MDP. Sensibly, the hard gridworld q-learner takes more time to keep running than the simple gridworld q-learner.

5.2.1 Easy Grid World Q-learning

Q-learning Parameters	Iterations	Time Taken	Reward	Steps Taken	Convergence
L0.1 Q0.0 E0.1	160	0.085	50.3100	47.2200	6.5800
L0.9 Q100.0 E0.3	296	0.0541	49.9600	48.4588	24.9595
L0.9 Q0.0 E0.3	134	0.0416	46.5600	47.4000	53.3545
L0.9 Q0.0 E0.5	17	0.0563	45.9200	50.9800	63.1121
L0.9 Q100.0 E0.5	585	0.1257	43.5340	47.1200	25.2423
L0.9 Q100.0 E0.1	354	0.0816	44.0600	48.4800	22.5748
L0.9 Q0.0 E0.1	950	0.1461	43.7200	26.2200	26.0067
L0.9 Q100.0 E0.5	876	0.1724	43.6400	25.3000	26.1176
L0.1 Q100.0 E0.1	204	0.0940	42.3800	52.1000	0.6510
L0.1 Q0.0 E0.3	943	0.1346	41.8345	26.56200	3.5511

Table 5: Easy Grid World Q-learning results

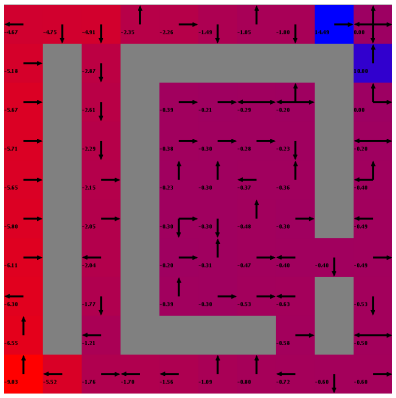

Figure 15: Q-Learning for Easy Grid World - 12 Iterations

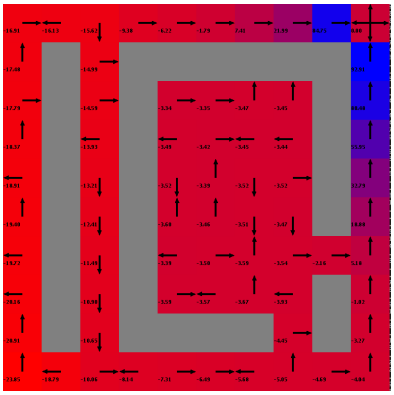

Figure 16: Q-Learning for Easy Grid World - 50 Iterations

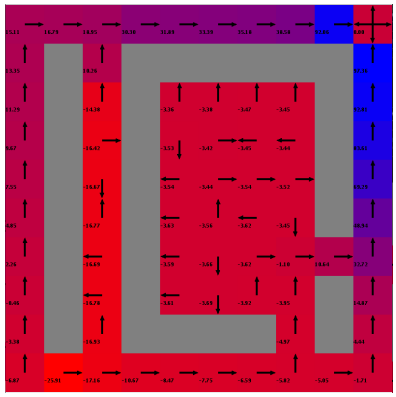

Figure 17: Q-Learning for Easy Grid World - 160 Iterations

5.2.2 Hard Grid World Q-learning

Q-learning Parameters	Iterations	Time Taken	Reward	Steps Taken	Convergence
L0.1 Q0.0 E0.1	2800	0.8533	20.1900	62.9800	2.1198
L0.1 Q100.0 E0.5	1882	0.8056	20.1070	63.8700	1.3662
L0.1 Q0.0 E0.3	2411	0.8169	19.5300	64.1900	2.9960
L0.1 Q0.0 E0.5	1177	0.5065	18.9700	62.9800	3.6562
L0.1 Q100.0 E0.3	2458	1.0309	18.4100	64.2200	1.8214
L0.1 Q100.0 E0.1	2685	0.9519	17.8000	63.1450	1.7738
L0.9 Q0.0 E0.3	1728	0.9165	5.7800	70.0500	64.5455
L0.9 Q0.0 E0.1	754	0.4340	-15.2900	63.1400	46.8360
L0.9 Q100.0 E0.5	1891	0.9692	-17.2500	89.3450	40.4389
L0.9 Q100.0 E0.1	2791	1.4653	-19.1800	62.6200	42.1840

Table 6: Hard Grid World Q-learning results

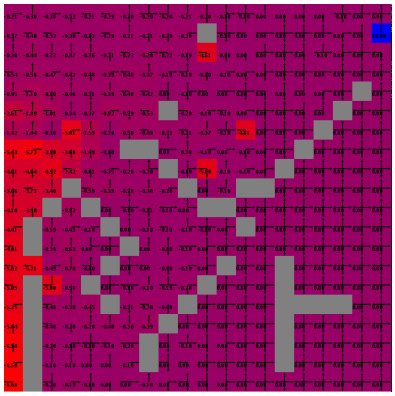

Figure 18: Q-Learning for Hard Grid World - 11 Iterations

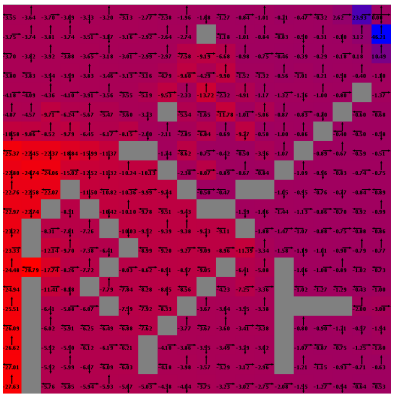

Figure 19: Q-Learning for Hard Grid World - 116 Iterations

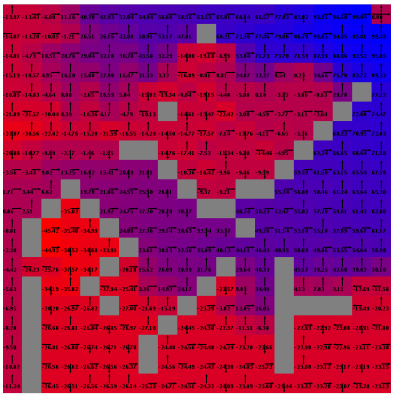

Figure 20: Q-Learning for Hard Grid World - 2800 Iterations

6 Conclusion

On comparison of the convergence properties for both MDP issues, it turns out to be exceptionally evident that policy iteration has the simplest time converging. While value iteration emphasis comes moderately near approach cycle, Q-learning has a substantially more difficult time.

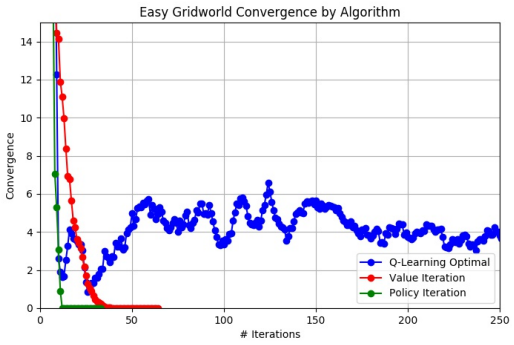

Figure 21: Easy GW Convergence

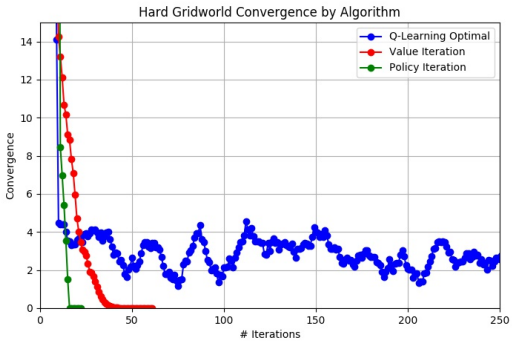

Figure 22: Hard GW Convergence

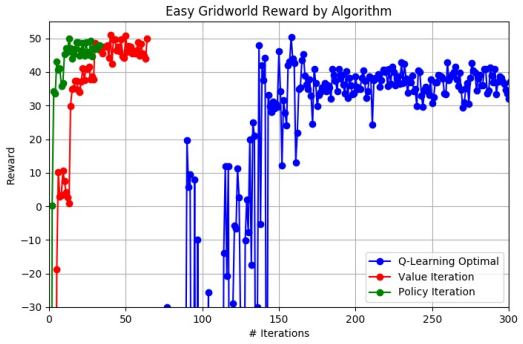

Figure 23: Easy GW reward

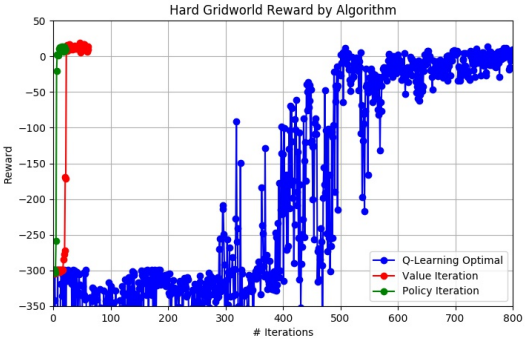

Figure 24: Hard GW Reward

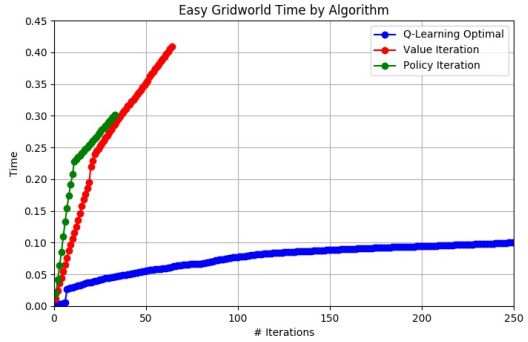

Figure 25: Easy Grid World Time

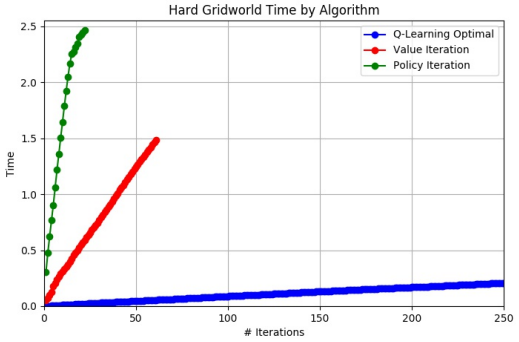

Figure 26: Hard Grid World Time

Since policy iteration begins with an random policy and afterward changes it for upgrades, it is helpless to converge rapidly due to not having much motivation to transform from it's present state. So also, while value iteration works in reverse, it creates varieties yet will definitely combine a little while later to an ideal arrangement as it is additionally helped by learning of the show. Q-learning, be that as it may, goes in totally blind and has a lot more noteworthy difficulty knowing whether the policy picked is optimal. With different choices to take at each state, it takes a lot more iteration to converge. As far as time important to run the calculation, Q-learning is by a long shot the quickest. Alternately, arrangement value is the slowest. Both of these announcements remain constant for both MDPs. Q-learning does not include any intricate math or computations, dissimilar to policy and value iteration. In this way, it can investigate and analyze its arrangements a lot quicker than the others. Q-learning can keep running in consistent time, while policy and values take steadily additional time. Reward analysis turns out to be the most fascinating examination over every one of the three calculations. Both approach policy and value iteration had the capacity to find maximal rewards rapidly as far as iterations, and had the capacity to keep that value rather reliable in spite of their investigations. Q-learning, in any case, battled significantly to find maximal reward polciies, and even once slanting towards maximal rewards, shifted significantly from value to policy. Polcy iteration having the upside of making improvement modifications to an underlying arrangement, is continually looking to boost its reward and is in this way ready to monotonically expand its reward return while achieving convergence. Value Iteration is not exactly as effective, as it's essentially working in reverse and attempting to find great alternatives en route, however regardless it performs or maybe well. Q-learning, then again, manages a lot of investigation and disclosure that makes it take a huge number of iteration before it finds arrangements that can guide to maximal prizes. Indeed, even once it begins performing great, it takes a long time to combine and is incredibly factor because of its propensity to even now endeavor to investigate new choices. In the process of thorough anlaysis, the two Markov Decision Processes were proposed and after that comprehended utilizing three different reinforcement learning algorithm: policy iteration,value iteration and Q-learning. The different algorithms were thoroughly analyzed, to touch base at different conditions where one might be more ideal than others. While Q-learning will in general take numerous iterations to converge, value and policy iterations do as such in many less cycles. Strikingly however, as far as crude time q-learning is rather a lot faster. If the model is known previously hand, value and policy will in general have the capacity to process the optimal policy significantly more suitably. If no model is known, Q-learning can deal with the circumstance better because of its capacity to investigate and find as it emphasizes.