Markov Decision process and Reinforcement Learning

mukund@gatech.edu

***Abstract*— The aim of this report is to explain the analysis of experiments with algorithms to solve Markov Decision Processes namely Value and Policy Iteration and one reinforcement learning algorithm. Experiments should tweak the parameters of algorithms in order to understand effect of parameters and to compare and contrast between the different algorithms.**

I. DATASETS

Maze:

This is a maze that involves finding the policy to reach a goal state through different possible ways. There may be obstacles/negative rewards in few paths namely -5, -6, -10 and -30 to denote two different kind of muddy paths, fire and . a pit draining into which will take you sometime to get back to the game.

1. Dataset 1 - Hard MDP

This is an instance of the above explained game with 15 * 11 = 165 states.

Since this is a big enough canvas, it gives us enough room to explore different possible reward combinations. This uses all kinds of rewards mentioned. Each reward exhibiting some effect towards finding the policy.

2. Dataset 2 - Easy MDP

This is an instance of the above explained game with 5 * 4 = 20 states.

Since this a very small canvas, this does not give enough room for exploration.

II. VALUE ITERATION

A. Hard MDP

Value iteration algorithm was run with parameters discount factor = 0.99, maxDelta = -1, and maxIterations was changed between 100 and 300.

Value Iteration performed really well and took only close to 20 iterations to come a stable state. Initial 9 iterations showed great levels of stochasticity after which there was a steady decrease in the number of steps taken. Reaching stable also meant that the algorithm has converged to the optimal policy.

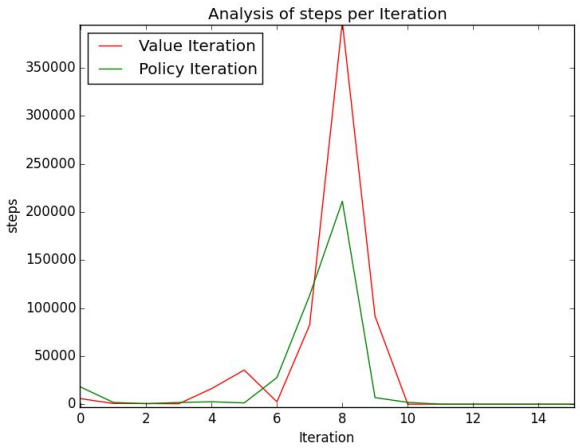


Fig 1.0

Above figure shows spike in the initial 10 iterations and this entering a stable state at about 10 iterations.

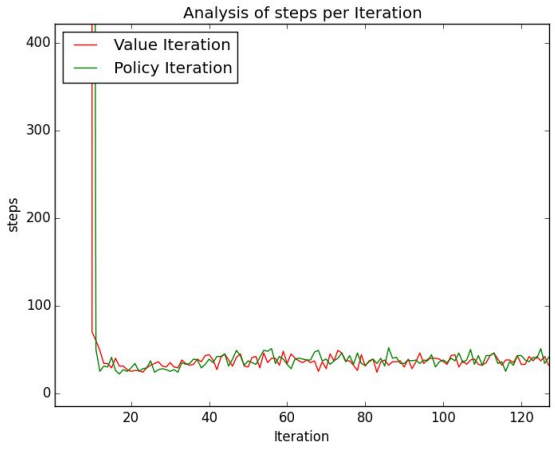


Fig 1.1

The above graph shows the stable levels of steps to goal.

Time taken per iteration increased linearly. Comparison between value and policy iteration is explained under the topic of policy iteration.

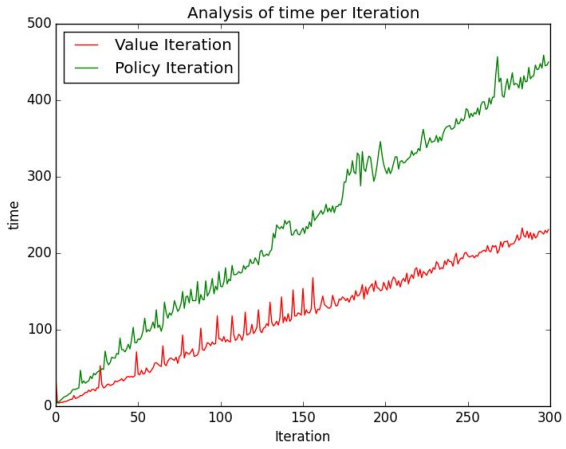
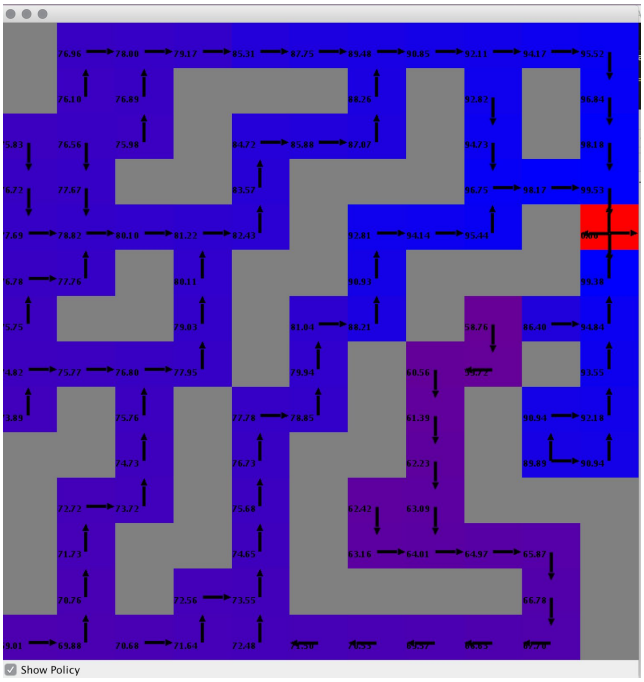


Fig 2

I. Optimal Policy Reasoning

Below graph shows the optimal policy as found by Value Iteration.



A closer look at the graph will give a better idea about the policy. In the maze seen above, there are several ways to reach the terminal state represented by a red square.

Defining (0, 0) as the left bottom cell, it can be seen that at (7, 9) is the brightest pink color. This reflects the presence of a pit with a reward of -30. As the algorithm has rightly found, anyone who comes close to the pit are asked to move away from it into another way. This is the expected behaviour in this case.

In the top left, at (0, 11) and (1, 11), the force is downwards repelling anyone coming in that way to go the other way round. This is because of the presence of a -5 reward at (4, 13). The effect of this has propagated towards the left path. The states closest to this negative reward are not affected since going the other way round will give them a more negative reward. So, at those states, it's best to go through the higher negative reward thus avoiding the round trip cost.

There is one other negative reward that is used in the maze that completely has no effect on the policy behaving as if it were the same as other neutral states. This is a negative reward of 6. This reward is in the middle of one of the paths. With a similar explanation as the previous one, since taking a roundabout path will incur more cost, the algorithm has decided to go through the negative cost.

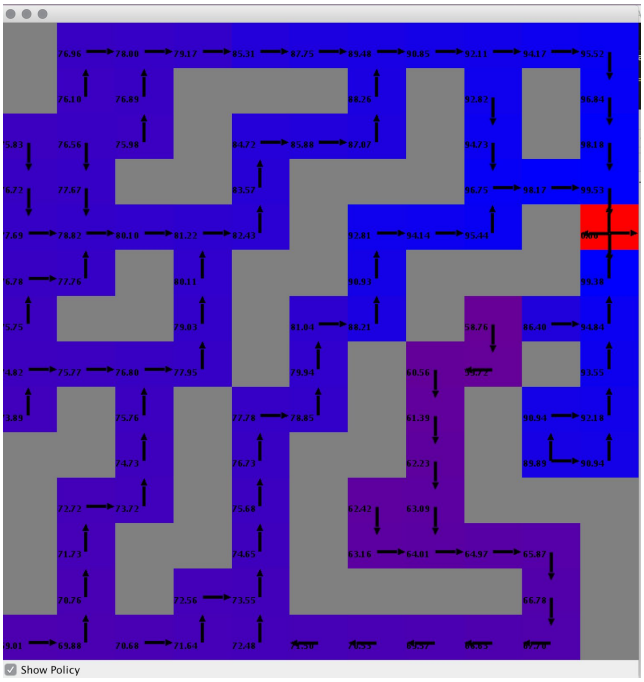
II. POLICY ITERATION

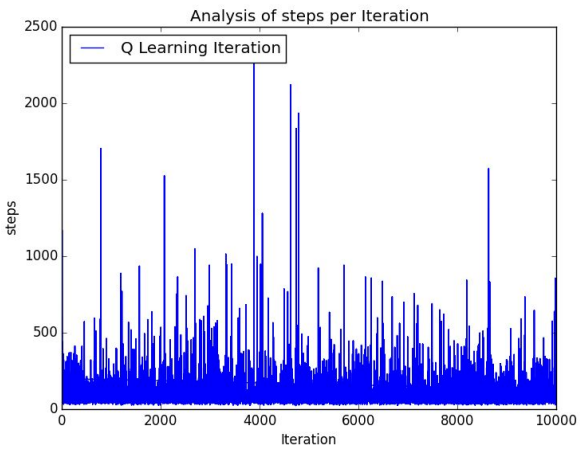
A. Hard MDP

Results of policy iteration on the hard problem is very similar to the results of value iteration. Initial 10 iterations were greatly stochastic after which they become stable close to 15 iterations. Similar to Value Iteration(refer fig 2), time taken linearly increases in this case as well, but in this case, it's 2x when compared to value iteration. For instance, at 37th iteration, value iteration took 33 seconds while policy iteration took 63 seconds.

This is inline with the theory that policy iteration takes longer per iteration but converges faster than value iteration. In this case, 20 iterations of value iteration to converge to 15 iterations of policy iteration to converge.

The optimal policy found was the same as that of the one found by Value Iteration.





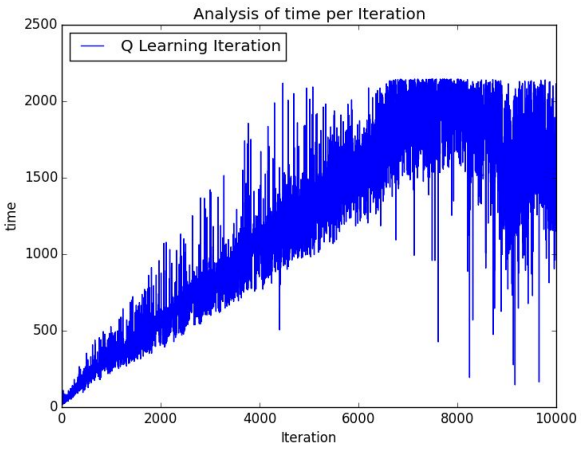
Above graph shows steps taken to reach goal state per iteration. It can also be seen that it did converge. 10000 iterations was not enough for the learning algorithm to converge to the optimal policy. While value iteration and policy iteration converged to the same policy, Q Learning was not able to converge to the same policy.

III. REINFORCEMENT LEARNING

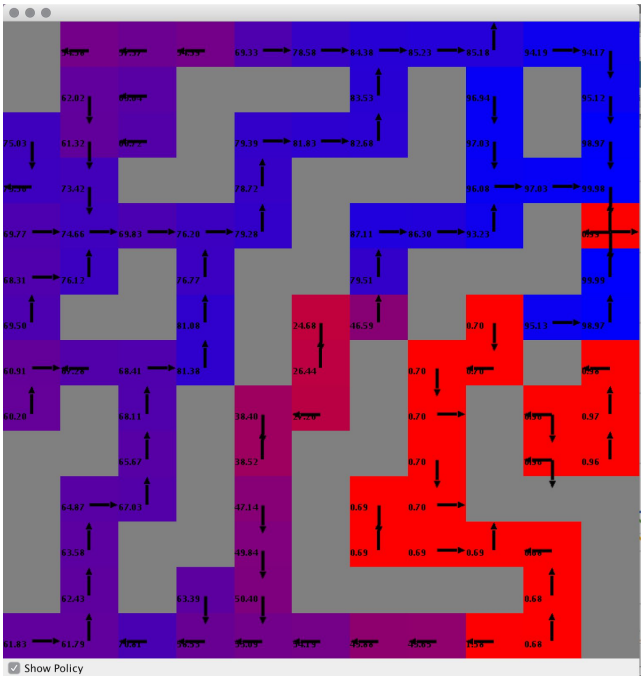
Choice of Reinforcement learning algorithm is Q Learning. The way it differs from Policy and value iteration is that, Q Learning does not know all the information about the surroundings.

A. Hard MDP

Q Learning was run with gamma, init, and learning rate set to 0.99. Experiment was run for 10000 iterations.



As can be seen from the above graph, time per iteration shows an erratic behavior. But the erratic behaviour internally follows a linear increase.



Above graph shows the derived optimal policy of Q Learning. It can be seen that the negative reward effects are maximized. Especially that of the pit which has aggravated the negative effect.