CSE 4633

Programming Assignment 4

Naveen Neelakandan (nn149)

Part1

In this assignment, we deal with the problem of MDPs where the outcome of an action is non-deterministic. The first algorithm implemented is form of dynamic programming called value iteration. For each state the V(S) value is updated based on the immediate reward and the expected future reward. When run long enough, the algorithm is guaranteed to converge. For our purposes, we stop the algorithm when the difference in the V(S) function is less than a certain small amount. It is noted that the algorithm is quick to converge to a reasonable amount (less than 50 iterations). The value of the discount factor is also noted to affect the results. A larger discount factor leads to almost equals importance to immediate and future rewards while a lower one prioritizes immediate rewards. As such the optimal policy varies slightly on adjusting the discount factor. The convergence of V(S) for each of the states in the grid is shown below.

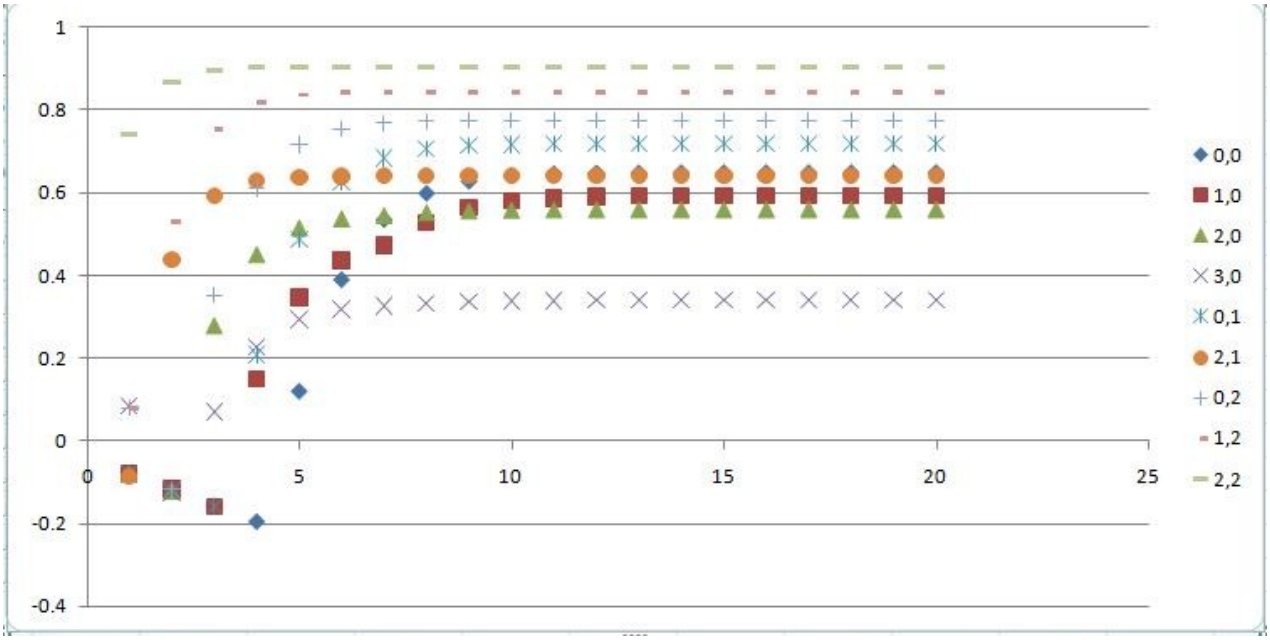

Figure 1 : Convergence of V(S) for each state in Value iteration --- Graph of V(S) vs No.of iterations

The second algorithm to be implemented is Q-learning. In the case of Q-Learning , our agent may not have information on the transition function and the reward for all the states. Instead, this is learned by exploring the environment. The Q-value for a state is the immediate reward for the action, plus the value of the best possible

state-action for the successor. The Q-value is thus updated on each state visited. The choice of which action to choose is unknown though. A greedy approach to choosing the action with the largest value may lead to the same states being picked over and over again, thus leading to less exploration of the state space. However, a random approach may lead to lots of unnecessary time spent in following sub-optimal paths. In this assignment, I elected to choose the next action randomly as the problem is a simple one and thus the issue of time is not as apparent. In a more complicated problem, we may begin with a random approach and the slowly shift to a greedy one when not much exploration is left. The learning rate alpha is also noted to affect the algorithm's performance. A higher value of alpha leads to faster convergence. The convergence of Q(S,A) for the optimal action A is shown below for each of the states.

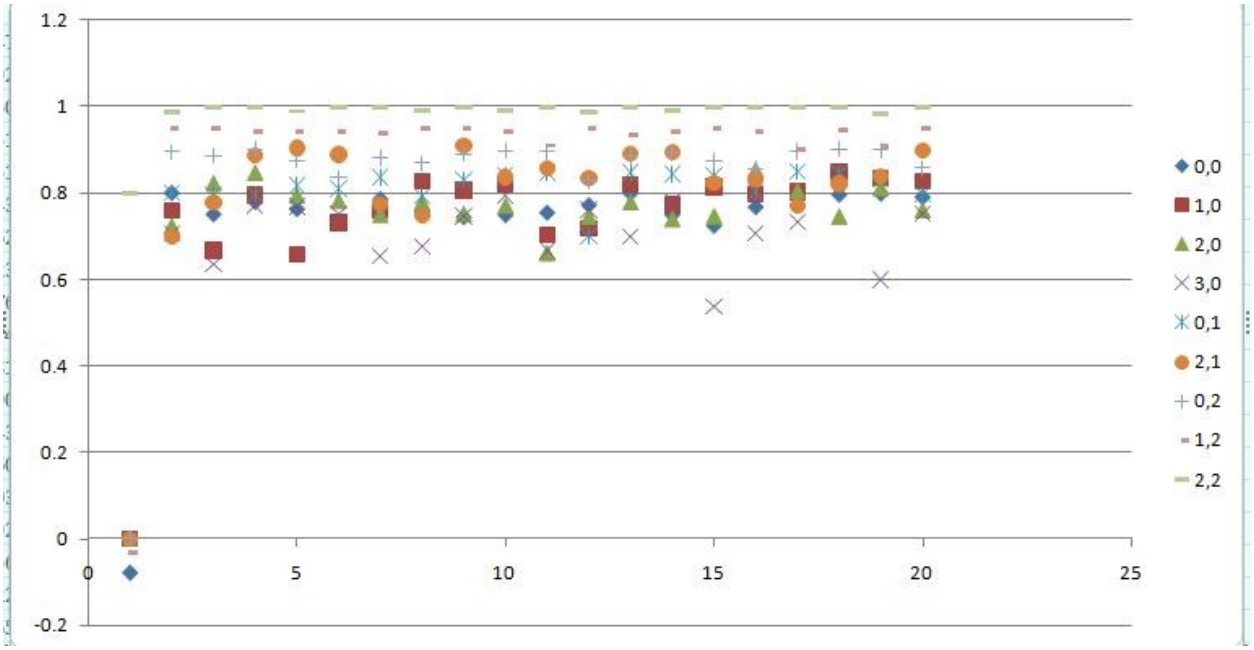

Figure 2 : Convergence of Q(S,A) for the optimal action A for each state in Q-learning --- Graph of Q(S,A) vs No.of iterations(100)