Reinforcement Learning: CW1

Haotian Wu (01864103)

October 30, 2023

Note that the

Q1: Dynamic programming

1

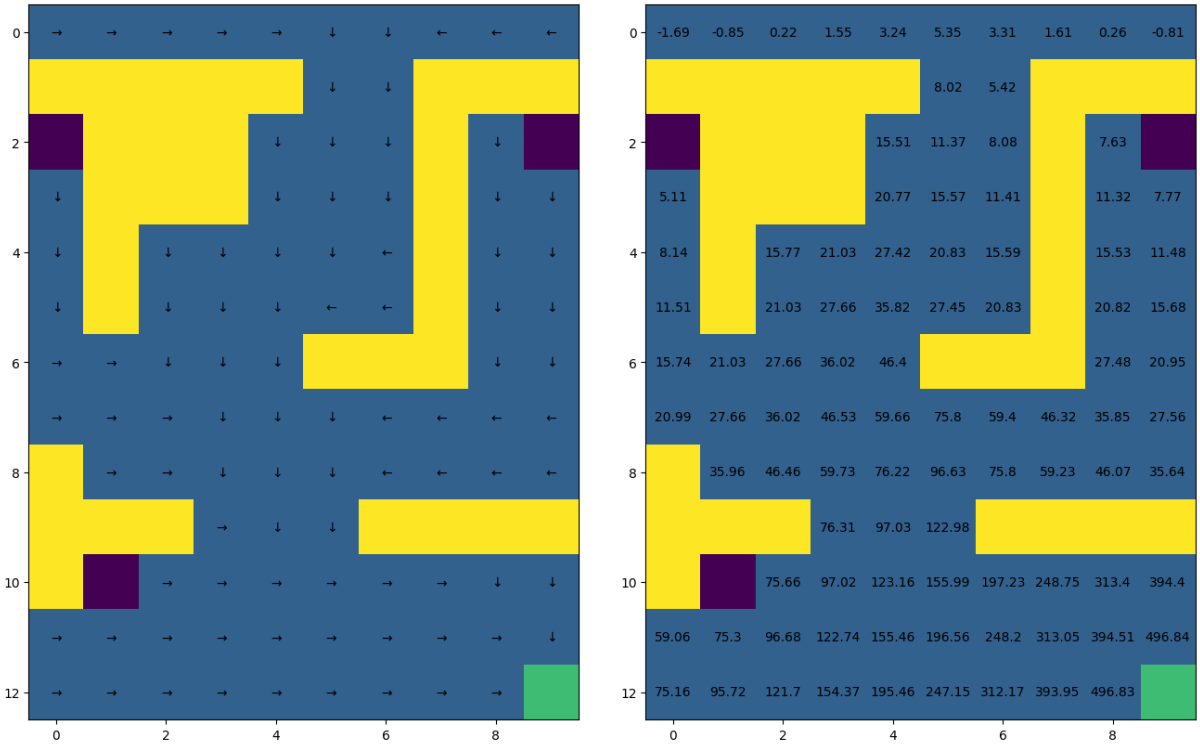
Policy Iteration was used here - policy iteration was chosen as it is in general faster. A small tolerance ($tolerance = 0.00001$) was set for the stopping condition of the policy evaluation step. This tolerance was chosen to be small enough as to confirm convergence of the policy evaluation. Especially as the rewards are to the nearest integer the tolerance is sufficiently small.

The Policy Iteration implementation is split into two stages: the policy evaluation and the policy improvement stages. Both of these are wrapped in a while loop that only terminates when the policy improvement stage does not update the policy (the flag $policy_stable$ is set to false if the policy is updated in policy improvement; otherwise is true and will terminate).

For the policy evaluation, Δ is initialised to $10 * \theta$ where $\theta = tolerance$. This ensures that the policy evaluation loop will be entered. There is a nested loop in Policy Evaluation, the outer one for looping through all available states and the inner for updating the value for the selected state, ie $V(s)$. The optimal policy given a state for the current iteration $\pi(s)$, is computed simply by finding the index with the largest value in the policy matrix corresponding to state. The current V is cached as each state requires the original $V(s')$ to be able update the current $V(s)$. $P_{ss'}^{\pi(s)}$ is the value from the transition matrix. The reward $R_{ss'}^{\pi(s)}$ is also computed given $\pi(s)$.

The optimal policy (action) is determined for each state

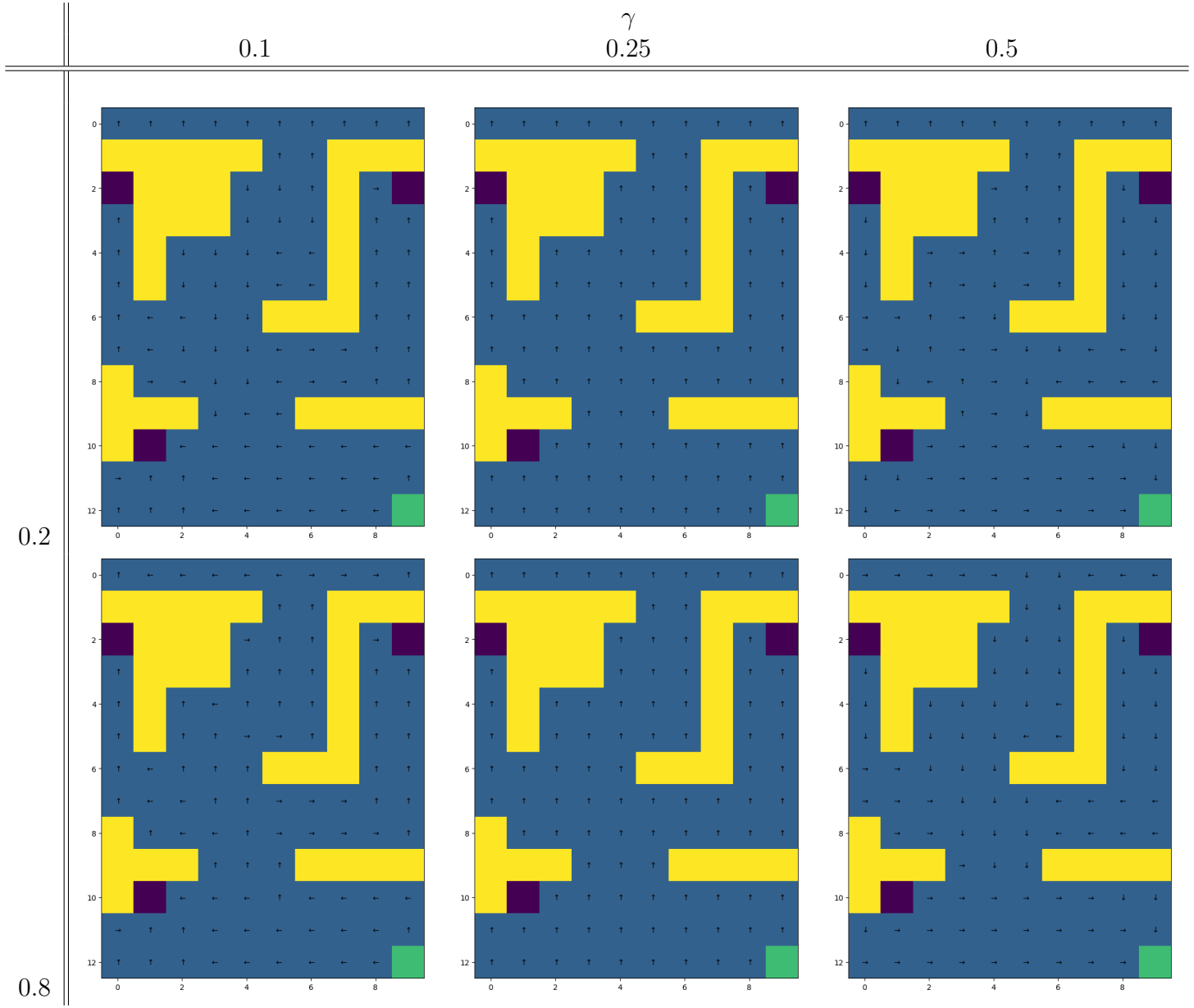
2

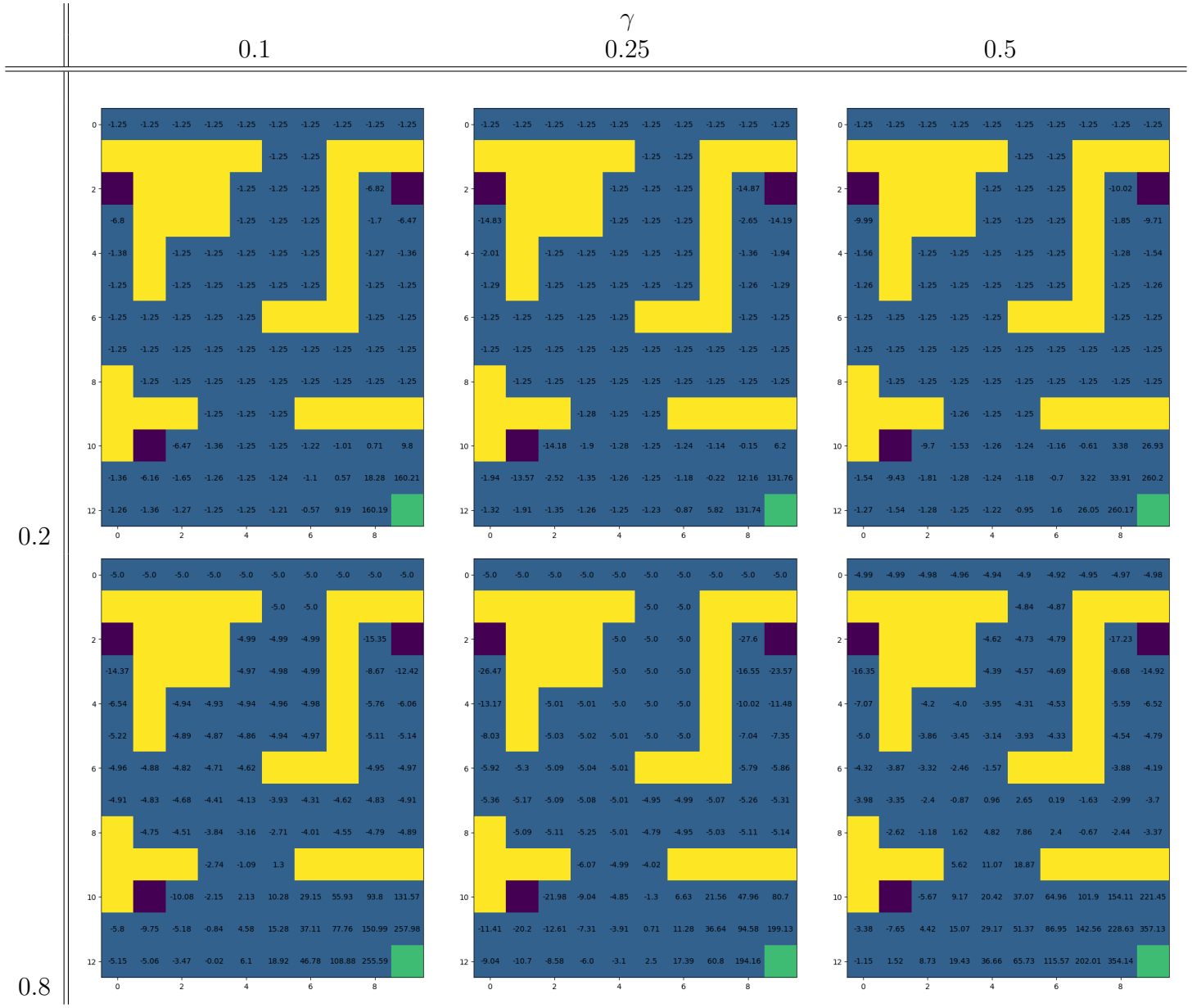


(a) Dynamic Programming Policy (b) Dynamic Programming Value FUnction

Graphical Representations of Dynamic Programming Results

3




γ
0.1
0.25
0.5
ε
0.2
0.8

When the probability $p = 0.25$, the transition matrix probability of success is 0.25 and the probability of going in another direction is also 0.25. This results in moving in another direction the same probability as the current optimal policy. Consider the policy update $\pi(s) \leftarrow argmax_a \sum_{s'} P^a_{ss'}[R^a_{ss'} + \gamma V(s')]$: the value $P^a_{ss'}$ will be biggest when the (When $p < 0.25$ this effect is exaggerated and the expected policy is not achieved.)