1 Effect of Reward Shifts

Prove that adding a constant $c$ to the reward of every state will change $V^*$ by exactly $c/(1 - \gamma)$. (Hint: Think about value iteration and recall some tricks for dealing with geometric series.)

2 Contraction of the Bellman Operator

In class, we showed that the Bellman operator without the max is a contraction in maximum norm, i.e., for

$$TV = R + \gamma PV,$$

assuming

$$||V_1 - V_2||_\infty = \epsilon,$$

then

$$||TV_1 - TV_2||_\infty \leq \gamma \epsilon$$

for discount factor $0 \leq \gamma < 1$. For this problem, we will consider the Bellman operator with max, i.e.,

$$T^*V(s) = R(s) + \gamma \max_a \sum_{s'} P(s'|s,a)V(s').$$

Prove that $T^*$ is a contraction in maximum norm. (We’re expecting a generalization of the proof in the class notes here.)

3 Divergence of Function Approximators

Consider a two state Markov chain with $P(s_2|s_1) = P(s_2|s_2) = 1$, i.e., a system which deterministically transitions to state $s_2$ and then stays in $s_2$ forever. For this problem, $R(s_1) = R(s_2) = 0$. Clearly $V^*(s_1) = V^*(s_2) = 0$ for all $\gamma$. Now consider a single feature $\phi(s_i) = i$ and an approximate value function of the form $\hat{V} = \phi w$. Obviously $w = 0$ is the best choice for $w$.

Now suppose we alternate value iteration and linear regression steps in hopes of finding the best $w$ automatically, starting with $i = 0$ and some initial guess $w_0$:

1. $V_{i+1} = R + \gamma P \phi w_i$
2. $w_{i+1} = \argmin_w ||V_{i+1} - \phi w||_2$
3. $i = i+1$
4. Goto 1

One might expect that alternating value iteration and regression in this way would lead to reasonable solution regardless of $\gamma$ and the initial choice of $w_0$, especially for such a simple problem as this. Demonstrate that this is not true in general. (Hint: This is easier than it sounds. Approach this by computing an expression for $w_{i+1}$ in terms of $w_i$ and $\gamma$.)
MDP and RL Implementation

For the following problems consider the simple robot navigation task on the grid shown below. The states are numbered 1 . . . 11. The black square indicates an obstacle. For each state, there are four possible actions: up, down, right and left, which behave according to the following rules:

- For all states except 4, any action that would move the robot off the grid, has the effect of returning the robot to its current position with probability 1, e.g., going up from state 1 returns the robot to state 1.
- For all states, any action that would move the robot into the black obstacle has the effect of returning the robot to its initial position with probability 1, e.g., going right from state 5 returns the robot to state 5 with probability 1.
- For all states except 4 and 6, any action that would move the robot into an empty square succeeds with probability 1, e.g., moving right from state 1 moves the robot to state 2.
- For state 6, the right and left actions behave as expected, but the floor is tilted slightly towards state 7, so the up and down actions succeed with probability only 0.95 and move the agent to state 7 with probability 0.05.
- For state 4, any action teleports the robot back to one of the other states (except 4) with probability 0.1 for each state.

Unless otherwise indicated, use a discount factor of $\gamma = 0.95$.

4 Creating the Model

Write a matlab file that sets up a reward vector and stores the transition probabilities.

5 Value Iteration

Write a small matlab program that solves for the optimal value function and policy using value iteration. Your program should take the transition model, reward, and discount factor as inputs, and return a value function and policy. (Note that since value iteration converges asymptotically, you will need to specify a termination condition. This is typically done by checking if the maximum change in the value function between iterations is less than some small constant.)

Please submit a print out of the resulting value function and policy.
6 Policy Iteration

Write a small matlab program that solves for the optimal value function and policy using policy iteration. Your
program should take the transition model, reward, and discount factor as inputs, and return a value function and
policy.

Please submit a print out of the resulting value function and policy.

7 Discounts

Demonstrate that a change in the optimal policy occurs when you change the discount factor, then explain, in
English, why this change has occurred.

8 Temporal Difference Learning

Implement temporal difference learning and use it to evaluate the optimal policy for this problem. Use a constant
learning rate of $\alpha = 0.05$ and an initial value function of 0. Plot the 2-norm of the difference between TD’s
value function and $V^*$ as a function of the number of steps in the environment. Don’t expect this difference
to go to 0 (or at least don’t expect it to stay there). Because of randomness in the transitions, the best you can
expect is for it hover around 0.