CS533– Reinforcement Learning

Instructor : Shashi Shekhar Jha (shashi@iitrpr.ac.in)

Home Assignment - 1 | Due on 02/10/2020 2400 Hrs | 60 Marks

Submission Instructions:

All submission is through google classroom in one zip file. In case you face any trouble with the submission, please contact the TA:

- Amanjot Kaur, amanjot.kaur@iitrpr.ac.in

or you can post your queries in the classroom.

Your submission must be your original work. Do not indulge in any kind of plagiarism or copying. Abide by the honour and integrity code to do your assignment.

Late submissions will attract penalties.

Penalty Policy: There will be a penalty of 5% for every 24 Hr delay in the submission. E.g. for first 24 Hr delay the penalty will be 10%, for submission with a delay of >24 Hr and < 48 Hr, the penalty will be 20% and so on.

You submission must include:

- A legible PDF document with all your answers to the assignment problems, stating the reasoning and output.
- A folder named as ‘code’ containing the scripts for the assignment along with the other necessary files to run your code.
- A README file explaining how to execute your code.

Naming Convention:

Name the ZIP file submission as follows: ***Name_rollnumber_HW1.zip*** E.g. if your name is ABC, roll number is 2017csx1234 and submission is for HW1 then you should name the zip file as: ABC_2017csx1234_HW1.zip

Part 1: Subjective Problems [35]

V $u \in V$ $v \in V$

Q.1. Let denote the space of all value functions. Let and be any two value functions such $v \geq u$ B $Bv \geq Bu$ that . Let denotes the bellman operator, show that . [6 points]

$\{v_0, v^1, v^2, \ldots\ldots, v^n, \ldots\}$

Q.2. Consider the value iteration algorithm. Let are the value functions in v^* $\forall n$ & $0 < \gamma < 1$,

successive iterations with as the optimal value function. Show that [6 points] $\| v^n - v^* \| \leq \gamma^n$

$1 - \gamma \| v^1 - v^0 \|$

Q.3. Given **S** states and **A** actions, derive the time complexity for : [10 points]

1. Value Iteration
2. Policy Iteration
3. Modified Policy Iteration where the policy evaluation step runs for only k iterations

$q_\pi(s, a) > v_\pi(s)$ π

Q.4. If , what can we say about policy , show with derivation. [3 points] γ

Q.5. Consider an MDP where the horizon is infinite i.e. there is no terminal state and the discount factor is .

V_π

A policy π in this MDP has a value function . Suppose we create a new MDP where the only difference is

$V^{new}{}_\pi$

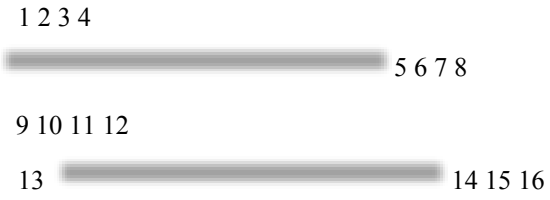
that all rewards have a constant **c** added to them. Derive an expression for the new value function V_π induced by π in the new MDP in terms of , c, and γ? [5 points]

Q.6. Consider the following grid environment. Starting from any unshaded square, you can move up, down, left, or right. Actions are deterministic and always succeed unless they will cause the agent to run into a wall. The thicker edges indicate walls, and attempting to move in the direction of a wall results in staying in the

R_G

same square. Taking any action from the green goal square earns a reward of and terminates the episode. R_D

Taking any action from the red square of death earns a reward of and terminates the episode. Otherwise, $R_S \in \{-1, 0, +1\}$ from every other square, taking any action is associated with a reward (even if the action R_G R_D results in the agent staying in the same square). Assume the discount factor $\gamma = 1$, $= +5$, and $= -5$ unless otherwise specified.



R_S

a) Choose the value of that would cause the optimal policy to return the shortest path to the green target R_S square (no. 12). Using this , find the optimal value for each square. [3 points]

V_π π

b) Lets denote the value function derived in (a) and the policy as . Suppose we are now in a new R_G R_D R_S π gridworld where all the rewards (, , and) have +2 added to them. Consider still following of the $V^{new}{}_\pi$ original gridworld, what will the new values be in the second gridworld? [2 points]

Part 2: Coding Problems [25 points]

For this part of the assignment, we will make use of the Open AI Gym environment. You can read about the Gym framework here: https://gym.openai.com/docs/

The gym framework bundles various environments to implement and compare various reinforcement learning algorithms. We will use one of the bundled environments called the Frozen Lake environment: https://gym.openai.com/envs/FrozenLake-v0/

You will implement the value iteration and policy iteration algorithms in your code for the Frozen Lake environment from OpenAI Gym. A custom version of this environment in the starter code zipped file. a) Read through `vi_and_pi.py` and implement `policy_evaluation, policy_improvement` and `policy_iteration`. The stopping criteria (defined as) is and $\max_s |V_{new}(s) - V_{old}(s)| < 10^{-3}$ $\gamma = 0.9$ $|V(s)| \forall s \in S$. The `policy_iteration` function should return the optimal value function and optimal policy. Provide a 3-D plot for after each `policy_evaluation` step until convergence. [10 points]

b) Implement `value_iteration` in `vi_and_pi.py`. The stopping criteria is and . The 10^{-3} $\gamma = 0.9$

`value_iteration` function should return the optimal value function and optimal policy. Provide a 3-$|V(s)| \forall s \in S$ D plot for for each iteration until convergence. [10 points]

(c) Run both methods (value iteration and policy iteration) on the Deterministic-4x4-FrozenLake-v0 and Stochastic-4x4-FrozenLake-v0 environments. In the second environment, the dynamics of the world are stochastic. How does stochasticity affect the number of iterations required, and the resulting policy? [5 points]