Assignment 3 Report

Saurabh Daptardar, Srikanth Kuthuru

October 1, 2017

1 Problem 1:

- Evaluating policies π_0, π_1, π_2 and optimal values V^* for $\gamma = 1$

	1	2	3	4	5
V^{π_0}	20	20	20	20	20
V^{π_1}	50	40	30	20	20
V^{π_2}	60	50	40	30	20
V^*	60	50	40	30	20

Table 1: Policy evaluation and optimal value function

- **Yes**, for $\gamma < 0.5$, π_0 is better than π_1 and π_2. For π_0 to be better we need to have $10 + \gamma 20 < 20$ i.e. $\gamma < 0.5$
- **No**. For π_1 to be better than π_0, we need to have $10 + \gamma 20 > 20$ and for it to be better than π_2 we must satisfy $10 + \gamma 20 < 20$, which is contradictory.
- **Yes**, for $\gamma > 0.5$, π_2 is better than π_0 and π_1. For π_0 to be better we need to have $10 + \gamma 20 > 20$ i.e. $\gamma > 0.5$

2 Problem 2:

MDP with 3 states that have rewards of -1,-2,0 is considered.

- Optimal policy at *state 1* is action b, because even if it tries to go to *state 2*, that state has the same transition probabilities as state 1, but more negative reward. So, going to *state 2* is not useful.
  Optimal policy at *state 2* is action a. Staying at *state 2* and trying to reach goal state might result in lot of negative points (-2 reward). Going to *state 1* first and then trying for goal state might be more fruitful.
- Given, initial policy for both states is action b. Initialize, Value function vector (V) to $[0, 0]$.

  Apply Policy evaluation step:

For *state 1*:

$$V_1 = 0.1(0+0) + 0.9(-1+V_1)$$
$$V_1 = -9$$

Similarly for *state 2*:

$$V_2 = 0.1(0+0) + 0.9(-2+V_2)$$
$$V_2 = -18$$

Next, do policy update:

$$\pi(s) = \underset{a \in Actions(s)}{\arg\max}\; Q(s,a)$$
$$Q(1,a) = 0.8(-2+V_2) + 0.2(-1+V_1) = -18$$
$$Q(1,b) = 0.9(-1+V_1) + 0.1(0+0) = -9$$

Therefore, action b should be preferred for *state 1*.
Similarly, for *state 2*:

$$Q(2,a) = 0.8(-1+V_1) + 0.2(-2+V_2) = -12$$
$$Q(2,b) = 0.9(-2+V_2) + 0.1(0+0) = -18$$

Therefore, action a should be prefered for *state 2*.
New policy: $\{1 : b, 2 : a\}$

Applying policy evaluation and policy iteration steps again, results in the same policy (as shown below).
Apply Policy evaluation step:

For *state 1*:

$$V_1 = 0.1(0+0) + 0.9(-1+V_1)$$
$$V_1 = -9$$

Similarly, for *state 2*:

$$V_2 = 0.8(-1+V_1) + 0.2(-2+V_2)$$
$$V_2 = -10.5$$

Next, do policy update:

$$\pi(s) = \underset{a \in Actions(s)}{\arg\max}\; Q(s,a)$$
$$Q(1,a) = 0.8(-2+V_2) + 0.2(-1+V_1) = -12$$
$$Q(1,b) = 0.9(-1+V_1) + 0.1(0+0) = -9$$

Therefore, action b should be preferred for *state 1*.
Similarly, for *state 2*:

$$Q(2,a) = 0.8(-1+V_1) + 0.2(-2+V_2) = -10.5$$
$$Q(2,b) = 0.9(-2+V_2) + 0.1(0+0) = -11.25$$

Therefore, action a should be preferred for *state 2*.
Therefore, this policy: $\{1 : b, 2 : a\}$ is optimal.

- If the initial policy for both states is action a, then policy evaluation step doesn't converge, or in other words, value function vector doesn't have a finite solution.
  Policy evaluation step gives the following equations:

  For *state 1*:

  $$V_1 = 0.8(-2+V_2) + 0.2(-1+V_1)$$
  $$0.8V_1 - 0.8V_2 = -1.8$$

  For *state 2*:

  $$V_2 = 0.8(-1+V_1) + 0.2(-2+V_2)$$
  $$0.8V_1 - 0.8V_2 = 1.2$$

  There is no finite solution for those linear equations.

- Including a discount factor $\gamma < 1$ works.
  Policy evaluation yields equations

  $$V_1 = 0.8(-2+\gamma V_2) + 0.2(-1+\gamma V_1)$$
  $$V_2 = 0.8(-1+\gamma V_1) + 0.2(-2+\gamma V_2)$$

  To evaluate policy we need to solve for V_1 and V_2:

  $$(1-0.2\gamma)V_1 - 0.8\gamma V_2 = -1.8$$
  $$(1-0.2\gamma)V_2 - 0.8\gamma V_1 = -1.2$$

  For $\gamma = 0.9$, $V_1 = -15.2$ and $V_2 = -14.8$ and the new policy: $\{1 : b, 2 : b\}$
  For $\gamma = 0.1$, $V_1 = -1.95$ and $V_2 = -1.38$ and the new policy: $\{1 : b, 2 : a\}$ (which happens to be the optimal policy)

3 Problem 3:

For subpart 3.1.6, there is a counter example provided in the code. Please evaluate that.

4 Problem 4:

The program to do value iteration is in file `prob4.py`. It does value iteration with convergence criteria as when policy stabilizes i.e. $\pi_{t+1} = \pi_t$ and prints out the result $V_t(s) \forall t \geq 0$ for each state for the given discount factors of $1, 0.75, 0.5, 0.1$. To get the output in text file run `python prob4.py >> output.txt`
The input MDP problem is specified in `input.txt` which has transition probabilities and rewards

- For discount = 1, the optimal policy is $\pi = \{1 : R, 2 : R, 3 : R\}$ and the result of value iteration is tabulated below

	1	2	3
$V_0(s)$	0.0	0.0	0.0
$V_1(s)$	8.0	16.0	7.0
$V_2(s)$	17.75	29.9375	17.875
$V_3(s)$	29.6640625	43.421875	30.90625

Table 2: For discount = 1

- For discount = 0.75, the optimal policy is $\pi = \{1 : C, 2 : R, 3 : R\}$ and the result of value iteration is tabulated below

	1	2	3
$V_0(s)$	0.0	0.0	0.0
$V_1(s)$	8.0	16.0	7.0
$V_2(s)$	15.3125	26.203125	14.40625

Table 3: For discount = 0.75

- For discount = 0.5, the optimal policy is $\pi = \{1 : C, 2 : R, 3 : C\}$ and the result of value iteration is tabulated below

	1	2	3
$V_0(s)$	0.0	0.0	0.0
$V_1(s)$	8.0	16.0	7.0
$V_2(s)$	12.875	22.46875	11.75

Table 4: For discount = 0.5

- For discount = 0.1, the optimal policy is $\pi = \{1 : C, 2 : C, 3 : C\}$ and the result of value iteration is tabulated below

	1	2	3
$V_0(s)$	0.0	0.0	0.0
$V_1(s)$	8.0	16.0	7.0

Table 5: For discount = 0.1