CS6700 : Reinforcement Learning
Written Assignment #1

Intro to RL, Bandits, DP — Deadline: 23 Feb 2020, 11:55 pm

Name: Rajan Kumar Soni — **Roll number:** cs18s038

- This is an individual assignment. Collaborations and discussions are strictly prohibited.
- Be precise with your explanations. Unnecessary verbosity will be penalized.
- Check the Moodle discussion forums regularly for updates regarding the assignment.
- Type your solutions in the provided LaTeXtemplate file.
- **Please start early.**

1. (2 marks) You have come across Median Elimination as an algorithm to get $(\epsilon, \delta)-$PAC bounds on the best arm in a bandit problem. At every round, half of the arms are removed by removing arms with return estimates below the median of all estimates. How would this work if we removed only one-fourth of the worst estimated arms instead? Attempt a derivation of the new sample complexity.

Solution: For the Median Elimination (ε, δ) algorithm we have that for every phase

$$\ell :$$

$$\mathbf{P}\left[\max_{j \in S_\ell} p_j \leq \max_{i \in S_{\ell+1}} p_i + \varepsilon_\ell\right] \geq 1 - \delta_\ell$$

Proof Without loss of generality we look at the first round and assume that p_1 is the reward of the best arm. We bound the failure probability by looking at the event $E_1 = \{\hat{p}_1 < p_1 - \varepsilon_1/2\}$, which is the case that the empirical estimate of the best arm is pessimistic.

Hoeffding's inequality: where n is the number of samples

$$\mathbf{P}\left[S_n \geq \mu + \varepsilon\right] \leq \exp(-2n\varepsilon^2) \quad \textit{and} \quad \mathbf{P}\left[S_n \leq \mu + \varepsilon\right] \geq \exp(-2n\varepsilon^2)$$

Since we sample sufficiently, we have that

$$\mathbf{P}\left[E_1\right] \leq \exp(-2n(\frac{\varepsilon_1}{2})^2) \tag{1}$$

In case E_1 does not hold, we calculate the probability that an arm j which is not an ε_1 -optimal arm is empirically better than the best arm.

$$\mathbf{P}\left[\hat{p}_j \geq \hat{p}_1 | \hat{p}_1 \geq p_1 - \varepsilon_1/2\right] \leq \mathbf{P}\left[\hat{p}_j \geq p_j + \varepsilon_1/2 | \hat{p}_1 \geq p_1 - \varepsilon_1/2\right] \leq \exp(-2n(\frac{\varepsilon_1}{2})^2)$$

Let # bad be the number of arms which are not ε_1 -optimal but are empirically better than the best arm, so that best arm at that round can be eliminated. We have that $\mathbb{E}\left[\#\text{ bad } |\hat{p}_1 \geq p_1 - \varepsilon_1/2\right] \leq k\exp(-2n(\frac{\varepsilon_1}{2})^2)$, where k is the number of arms. Next we apply Markov inequality to obtain,

$$\mathbf{P}\left[\#\text{ bad } \geq 3k/4 | \hat{p}_1 \geq p_1 - \varepsilon_1/2\right] \leq \frac{k\exp(-2n(\frac{\varepsilon_1}{2})^2)}{3k/4} \tag{2}$$

Using the union bound we add (1) and (2) gives us that the probability of failure is bounded by δ_1.

$$\frac{k\exp(-2n(\frac{\varepsilon_1}{2})^2)}{3k/4} + \exp(-2n(\frac{\varepsilon_1}{2})^2 \leq \delta_1 \tag{3}$$

Upon solving we got

$$n \geq \frac{2}{\varepsilon_1^2}\log(\frac{7}{3\delta_1})$$

NOTE: From here notation has changed, for number of arms nwill be used instead of k The number of arm samples in the ℓ -th round is $2n_\ell \log\left(7/\delta_\ell 3\right)/\varepsilon_\ell^2$. By definition we have that

1. $\delta_1 = \delta/2; \delta_\ell = \delta_{\ell-1}/2 = \delta/2^\ell$
2. $n_1 = n; n_\ell = n_{\ell-1} * 3/4 = (\frac{3}{4})^{\ell-1} * n$
3. $\varepsilon_1 = \varepsilon/8; \quad \varepsilon_\ell = \frac{7}{8}\varepsilon_{\ell-1} = \left(\frac{7}{8}\right)^{\ell-1}\varepsilon/8$

$$\sum_{\ell=1}^{\log_{\frac{4}{3}}(n)} \frac{n_\ell \log\left(7/3\delta_\ell\right)}{\left(\varepsilon_\ell\right)^2/2} = 2\sum_{\ell=1}^{\log_{\frac{4}{3}}(n)} \frac{(\frac{3}{4})^{\ell-1} * n\log\left((7*2^\ell)/3\delta\right)}{\left(\left(\frac{7}{8}\right)^{\ell-1}\varepsilon/8\right)^2}$$

$$= \frac{128n}{\varepsilon^2}\sum_{\ell=1}^{\log_{\frac{4}{3}}(n)} \left(\frac{48}{49}\right)^{\ell-1}\log((7*2^\ell)/3\delta)$$

$$= \frac{128n}{\varepsilon^2}\sum_{\ell=1}^{\log_{\frac{4}{3}}(n)} \left(\frac{48}{49}\right)^{\ell-1}(\log(\frac{1}{\delta}) + \ell\log(2) + \log(\frac{7}{3}))$$

$$\leq 128\frac{n\log(1/\delta)}{\varepsilon^2}\sum_{\ell=1}^{\infty}\left(\frac{48}{49}\right)^{\ell-1}(\ell C' + C) = O\left(\frac{n\log(1/\delta)}{\varepsilon^2}\right)$$

2. (3 marks) Consider a bandit problem in which you know the set of expected payoffs for pulling various arms, but you do not know which arm maps to which expected payoff. For example, consider a 5 arm bandit problem and you know that the arms 1 through 5 have payoffs 3.1, 2.3, 4.6, 1.2, 0.9, but not necessarily in that order. Can you design a regret minimizing algorithm that will achieve better bounds than UCB? What makes you believe that it is possible? What parts of the analysis of UCB will you modify to achieve better bounds?

Solution: Yes,I can design a regret minimizing algorithm that will achieve better bounds than UCB.

Since we know that true expected payoffs. The task here is mapping of true payoffs to arm. I think it is possible because , we know the true values. As we start sampling and finding the average then bounds for respective arms will also start shrinking.many of the true expectation will start coming within the bound. As soon as all true payoffs will be nearest to some emprical expectation with good confidence we can stop sampling the arm.

Below is the simple modification:- As it does not assume any distribution over rewards. As we know that in UCB initially each arm are selected and then at each step that arm is selected whose $(Q(j) + \sqrt{\frac{2 \ln n}{n_j}})$, is high. Her first term is the empirical expectation and second term is the uncertainty associated with it.

A better regret minimizing algorithm is UCB with some modification where action can be chosen according to $(Q(j) + \sqrt{\frac{2 \ln n}{n_j}} \Delta j)$ where $\Delta j = q_*(a^*) - q * (a_i)$ where a_i is a suboptimal action. We can say if uncertainity in USB falls below $0.5\,(q_*(a^*) - q * (a_i)) = 0.5(4.6 - 3.1) = 0.75$ then the action with highest estimate of Q will be the optimum action. we are confident with is arm as most worst can be$< 3.1 + 0.75 = 3.85$ and optimum arm is greater then 4.6-.75=3.85

3. (3 marks) Suppose you face a 2-armed bandit task whose true action values change randomly from time step to time step. Specifically, suppose that, for any time step, the true values of actions 1 and 2 are respectively 0.1 and 0.2 with probability 0.5 (case A), and 0.9 and 0.8 with probability 0.5 (case B).

   (a) (1 mark) If you are not able to tell which case you face at any step, what is the best expectation of success you can achieve and how should you behave to achieve it?

   Solution:

So assuming a uniform distribution over which type of case we are given and where we play many games is given by

$$0.5(0.5) + 0.5(0.5) = 0.5$$

Since here we are not told that which case we will face at given point of time. As it is given both case are equally probable. In general if we knew which case we were facing we would like to pull different arms (the optimal play would be to pull the second arm in the case of A instance and pull the first arm in the case of a case B instance).So here we can find the expected value for each action.

$$E[Action1] = 0.5(0.1) + 0.5(0.9) = 0.5$$
$$E[Action2] = 0.5(0.2) + 0.5(0.8) = 0.5$$

Since here we can see that expected reward for both actions are same. So we can choose any of the action through out the run.To be able to achieve this optimal behavior, we could just use one of the standard stationary approaches, such as **UCB** with incremental **Q**

(b) (2 marks) Now suppose that on each step you are told whether you are facing case A or case B (although you still don't know the true action values). This is an associative search task. What is the best expectation of success you can achieve in this task, and how should you behave to achieve it?

Solution:

In this case if we told that which we are facing then it be good for us.As we know that case A and case B have different action values for the same set of actions. As we have already told that which case I am facing we can put different set of **Q** values for different case and upate them independently according to case. We can any algo like UCB then we will have best actions in both case and as we know that both cases are equally probable.

$$Best \;\; expectation \;\; of \;\; success = 0.9 * 0.5 + 0.2 * 0.5 = .5$$

4. (5 marks) Many tic-tac-toe positions appear different but are really the same because of symmetries.

   (a) (2 marks) How might we amend the learning process described above to take advantage of this? In what ways would this change improve the learning process?

Solution:
we might amend the learning process described below to take advantage of this-we can divide the state spaces into classes according to the symmetries. One way to handle this, we can use the 4-axis symmetry of tic-tac-toe to fold the board in one quarter.

We can say mainly two ways it will improve the process:

- **Time taken for convergence:** For updation of each state value, we can use the symmetric states value to upcate it, which can lead us faster convergence.
- **Space required to store the values:** As we are using same values for symmetric states so space requirement will be decreased.

(b) (1 mark) Suppose the opponent did not take advantage of symmetries. In that case, should we? Is it true, then, that symmetrically equivalent positions should necessarily have the same value?

Solution:
If the opponent does not take advantage of symmetric states, it means that state values and action values will be different for symmetric states. This means that those states are not really equivalent for our purposes and we should exploit this fact to improve our strategy against the opponent.

Example: Lets consider a game where two player A and B playing the game. Suppose player A don't use the symmetries of the tic-tac-toe and it takes different action on symmetric states. consider two states S_1 and S_2. Assume player A is weak player then for stateS_1 it makes good move and for state S_2 it makes bad move. If we would have maintained separate values for those two states then we can exploit it.

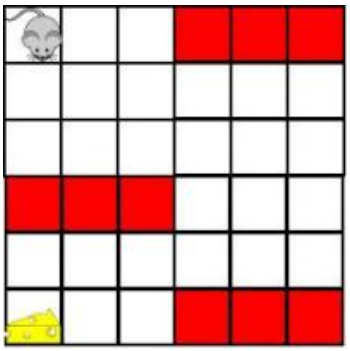
It is not true, that symmetrically equivalent positions should necessarily have the same value. Symmetrically equivalent positions don't need to hold the same value always in a multi-player game

(c) (2 marks) Suppose, instead of playing against a random opponent, the reinforcement learning algorithm described above played against itself, with both sides learning. What do you think would happen in this case? Would it learn a different policy for selecting moves?

Solution:
Since agent is playing itself it means we can say that when agent start playing it was using random policy. Since agent will improve itself because it will exploit its own bad moves. in this way agent keep on improving and will learn optimal policy.
In both cases agent try to learn optimal policy, there can be more than on optimal policy, so there is no need that policy will same.
If we don't talk about optimal policy then agent policy will depend on opponent policy, so it can be different form the policy when it is playing against itself.

5. (1 mark) Ego-centric representations are based on an agent's current position in the world. In a sense the agent says, I don't care where I am, but I am only worried about the position of the objects in the world relative to me. You could think of the agent as being at the origin always. Comment on the suitability (advantages and disadvantages) of using an ego-centric representation in RL.

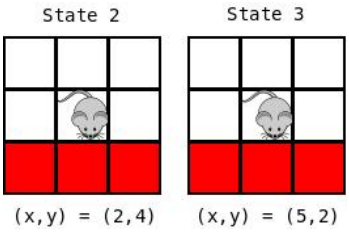
Solution: Taking the example of mouse and cheese cliff world.



Disdvantages of ego-centric:

- With this type of learning, agents quickly develop such a behaviour which is immediately beneficial.It uses this behaviorto avoid running off cliffs and to move towards cheese when it's in sight.
- The agent can even learn to move towards cheese that's outside of the range of its sensors in simple maps, essentially by learning heuristics like 'always moving right leads to reward.
- It will also work better when environment is dynamic.

Advantages of ego-centric:



- But ego-centric can fail if there are several areas in the map where the immediate environment is the same, as in the case above with states 2 and 3.
- If the environment is not dynamic then calculating the values every time will increase computing time.

6. (2 marks) Consider a general MDP with a discount factor of γ. For this case assume that the horizon is infinite. Let π be a policy and V^π be the corresponding value function. Now suppose we have a new MDP where the only difference is that all rewards have a constant c added to them. Derive the new value function V^π_{new} in terms of V^π, c and γ.

Solution: As we know that return at time for any state is given as

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots$
$= R_{t+1} + \gamma \left(R_{t+2} + \gamma R_{t+3} + \gamma^2 R_{t+4} + \cdots \right)$ which is a summation of reward mul-
$= R_{t+1} + \gamma G_{t+1}$
tipled beγ. So by adding a constant c in every reward we have,

$$G^*_t \doteq (R_{t+1} + c) + \gamma(R_{t+2} + c) + \gamma^2(R_{t+3} + c) + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} + c \sum_{k=0}^{\infty} \gamma^k$$

$$G^*_t \doteq G_t + \textstyle\sum_{k=0}^{\infty} \gamma^k C = G_t + \frac{C}{1-\gamma}$$
$$v^{new}_\pi(s) \doteq \mathbb{E}\left[G^*_t | S_t = s\right] = \mathbb{E}\left[G_t + \frac{C}{1-\gamma} | S_t = s\right] = \mathbb{E}\left[G_t | S_t = s\right] + \frac{C}{1-\gamma}$$
$$= v_\pi + \frac{C}{1-\gamma}$$

From the last step we concluded that all the state value increases by same amount so it does not effect relative of all states.

7. (4 marks) An ϵ-soft policy for a MDP with state set $\mathcal{S}$ and action set $\mathcal{A}$ is any policy

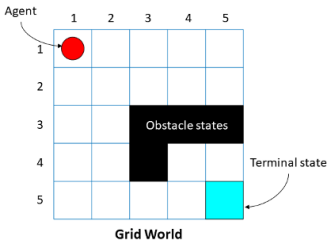
that satisfies

$$\forall a \in \mathcal{A}, \forall s \in \mathcal{S} : \pi(a|s) \geq \frac{\epsilon}{|\mathcal{A}|}$$

Design a stochastic gridworld where a deterministic policy will produce the same trajectories as a ϵ-soft policy in a deterministic gridworld. In other words, for every trajectory under the same policy, the probability of seeing it in each of the worlds is the same. By the same policy I mean that in the stochastic gridworld, you have a deterministic policy and in the deterministic gridworld, you use the same policy, except for ϵ fraction of the actions, which you choose uniformly randomly.

(a) (2 marks) Give the complete specification of the world.

Solution:



Since it is a grid world, So task are episodic so no requirement of discounting factor.
Also reward at every state is -1.
Reward function is: $R(s^{'}, a, s) = -1$ for all states except for goal **Action set:** = { Left, Right, Up, Down}
For deterministic world: Since $\epsilon-$soft policy is used Let say,my current policy for some state is
$p(Right/s) = 1 - \epsilon$
$p(Left/s) = \epsilon/3$
$p(Up/s) = \epsilon/3$
$p(Down/s) = \epsilon/3$
For stochastic world: Since here policy is deterministic
for some state s
$p(a = Right/s) = 1$
$p(s^{'} = (Right\ side\ of\ s)/s, a = \ Right)) = 1 - \epsilon$
$p(s^{'} = (Left\ side\ of\ s)/s, a = \ Right)) = \epsilon/3$
$p(s^{'} = (Up\ side\ of\ s)/s, a = \ Right)) = \epsilon/3$
$p(s^{'} = (Down\ side\ of\ s)/s, a = \ Right)) = \epsilon/3$

If this is the scenario satisfied then above stochastic gridworld where a deterministic policy will produce the same trajectories as $\epsilon - soft$ policy in a deterministic gridworld.

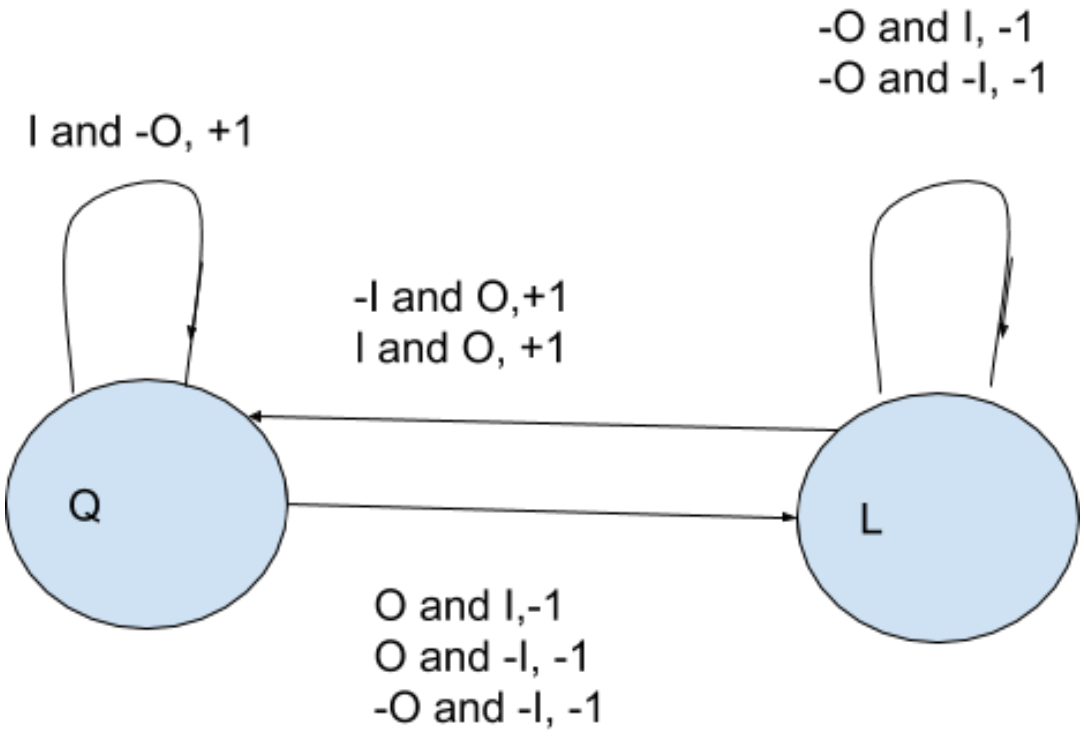
(b) (2 marks) Will SARSA on the two worlds converge to the same policy? Justify.

> **Solution:** If reward does not depend on action and it only depends on state s_t and s_{t+1} then thew will converge to the same policy.

8. (7 marks) You receive the following letter:
Dear Friend, Some time ago, I bought this old house, but found it to be haunted by ghostly sardonic laughter. As a result it is hardly habitable. There is hope, however, for by actual testing I have found that this haunting is subject to certain laws, obscure but infallible, and that the laughter can be affected by my playing the organ or burning incense. In each minute, the laughter occurs or not, it shows no degree. What it will do during the ensuing minute depends, in the following exact way, on what has been happening during the preceding minute: Whenever there is laughter, it will continue in the succeeding minute unless I play the organ, in which case it will stop. But continuing to play the organ does not keep the house quiet. I notice, however, that whenever I burn incense when the house is quiet and do not play the organ it remains quiet for the next minute. At this minute of writing, the laughter is going on. Please tell me what manipulations of incense and organ I should make to get that house quiet, and to keep it so.
Sincerely,
At Wits End

(a) (3 marks) Formulate this problem as an MDP (for the sake of uniformity, formulate it as a continuing discounted problem, with $\gamma = 0.9$. Let the reward be +1 on any transition into the silent state, and -1 on any transition into the laughing state.) Explicitly give the state set, action sets, state transition, and reward function.

> **Solution:** State set: $\{L, Q\}$, where L indicates that there is laughter in the room, and Q indicates that the room is quiet.
> Action set: $\{O \wedge I, O \wedge \neg I, -O \wedge I, \neg O \wedge \neg I,$ where O corresponds to playing the organ, and I corresponds to burning incense.
> Reward Function:
> $\mathrm{R(Q, O, L)} = -1, \mathrm{R(L, O, Q)} = +1, \mathrm{R(Q, I, Q)} = +1, \mathrm{R(L, I, L)} = -1$



old sate	Action taken	next state	reward
(L	$\mathrm{O} \wedge \mathrm{I}$	Q	$+1$
(L	$\mathrm{O} \wedge \neg \mathrm{I}$	Q	$+1$
(L	$\neg O \wedge \neg I$	L	$-1)$
(L	$\neg O \wedge I$	L	$-1)$
(Q	$\mathrm{O} \wedge \neg I$	L	$-1)$
(Q	$\neg O \wedge \neg I$	L	$-1)$
(Q	$\neg O \wedge I$	Q	$+1)$
(Q	$O \wedge I$	L	$-1)$

(b) (2 marks) Starting with simple policy of **always** burning incense, and not playing organ, perform a couple of policy iterations.

Solution:
Using equations for policy evaluation

$$\begin{aligned} v_\pi(s) &= \sum_a \pi(a|s) \sum_{r,s'} p\left(r, s'|s, a\right)\left[r + \gamma v_\pi\left(s'\right)\right] \\ &= \mathbb{E}_\pi\left[R_t + \gamma v'_\pi\left(S_{t+1}\right)|S_t = s\right] \end{aligned} \tag{4}$$

Using equation for policy improvement

$$\pi'(s) \leftarrow \underset{a}{\text{argmax}} \overbrace{\sum_{r,s'} p\left(r, s'|s, a\right)\left[r + \gamma v_\pi\left(s'\right)\right]}^{q_\pi(s,a)} \tag{5}$$

lets $\gamma = .5$ Initialising policy
$\pi_0(A = (\text{O} \wedge \text{I})/S = L) = 0,$
$\pi_0(A = (\neg\text{O} \wedge \text{I})/S = L) = 1,$
$\pi_0(A = (\text{O} \wedge \neg\text{I})/S = L) = 0,$
$\pi_0(A = (\neg\text{O} \wedge \neg\text{I})/S = L) = 0$
$\pi_0(A = (\text{O} \wedge \text{I})/S = Q) = 0,$
$\pi_0(A = (\neg\text{O} \wedge \text{I})/S = Q) = 1,$
$\pi_0(A = (\text{O} \wedge \neg\text{I})/S = Q) = 0,$
$\pi_0(A = (\neg\text{O} \wedge \neg\text{I})/S = Q) = 0$
Initialising value of states $V_{\pi_0}(L) = 0$ and $V_{\pi_0}(Q) = 0$

As the policy is very sparce using equation (4) Using Bellman equation, we get $V_{\pi_0}(Q) = 1 + \gamma V_{\pi_0}(Q)$ and $V_{\pi_0}(L) = -1 + \gamma V_{\pi_0}(L)$ $V_{\pi_0}(Q) = 2, V_{\pi_0}(L) = -2$
policy improvement:

$$\pi_1(Q) = \underset{a}{\text{argmax}} \begin{cases} +1 + 2 * .5 & ,(\neg\text{O} \wedge \text{I}) \\ -1 - 2 * .5 & ,(\text{O} \wedge \text{I}) \\ -1 - 2 * .5 & ,(\text{O} \wedge \neg\text{I}) \\ -1 - 2 * .5 & ,(\neg\text{O} \wedge \neg\text{I}) \end{cases}$$

$\pi_1(Q) = (\neg\text{O} \wedge \text{I})$

$$\pi_1(L) = \underset{a}{\text{argmax}} \begin{cases} -1 - 2 * .5 & ,(\neg\text{O} \wedge \text{I}) \\ +1 + 2 * .5 & ,(\text{O} \wedge \text{I}) \\ +1 + 2 * .5 & ,(\text{O} \wedge \neg\text{I}) \\ -1 - 2 * .5 & ,(\neg\text{O} \wedge \neg\text{I}) \end{cases}$$

$\pi_1(L) = (\text{O} \wedge \text{I}), (\text{O} \wedge \neg\text{I})$

selecting any acton from the above $\pi_1(L)$

Our new policy
$\pi_1(A = (\text{O} \wedge \text{I})/S = L) = 1,$
$\pi_2(A = (\neg\text{O} \wedge \text{I})/S = L) = 0,$
$\pi_2(A = (\text{O} \wedge \neg\text{I})/S = L) = 0,$

$\pi_1(A = (\neg \mathrm{O} \wedge \neg \mathrm{I})/S = L) = 0$
$\pi_1(A = (\mathrm{O} \wedge \mathrm{I})/S = Q) = 0,$
$\pi_1(A = (\neg \mathrm{O} \wedge \mathrm{I})/S = Q) = 1,$
$\pi_1(A = (\mathrm{O} \wedge \neg \mathrm{I})/S = Q) = 0,$
$\pi_1(A = (\neg \mathrm{O} \wedge \neg \mathrm{I})/S = Q) = 0$

policy evaluation: Applying again equation 4 and using Bellman equation
Still here policy is vary sparse
$V_{\pi_1}(L) = +1 + .5 * 2 = 2$ and $V_{\pi_1}(Q) = +1 + .5 * 2 = 2,$
Here V values dose not changed it is already converged, no change in value function so policy will also not change

(c) (2 marks) Finally, what is your advice to "At Wits End"?

Solution: My advice will be if there is laughter, play the organ; if room is quite, do not play the organ and burn incense

9. (4 marks) Consider the task of controlling a system when the control actions are delayed. The control agent takes an action on observing the state at time t. The action is applied to the system at time $t + \tau$. The agent receives a reward at each time step.

(a) (2 marks)What is an appropriate notion of return for this task?

Solution: As we know that return at time for any state is given as

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$
$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots$$

$$G_t = R_{t+1} + \gamma \left(R_{t+2} + \gamma R_{t+3} + \gamma^2 R_{t+4} + \cdots\right) = R_{t+1} + \gamma G_{t+1}$$

As the action getting delayed:

$$G_t \doteq R_{t+1+\tau} + \gamma R_{t+2+2\tau} + \gamma^2 R_{t+3+3\tau} + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1+\tau(k+1)}$$
$$G_t \doteq R_{t+1+\tau} + \gamma R_{t+2+2\tau} + \gamma^2 R_{t+3+3\tau} + \gamma^3 R_{t+4+4\tau} + \cdots$$

$$G_t = R_{t+1+\tau} + \gamma \left(R_{t+2+2\tau} + \gamma R_{t+3+3\tau} + \gamma^2 R_{t+4+4\tau} + \cdots\right) = R_{t+1+\tau} + \gamma G_{t+1+\tau}$$

(b) (2 marks) Give the TD(0) backup equation for estimating the value function of a given policy.

Solution:

$$\mathrm{V}\left(\mathrm{S_t}\right) \leftarrow \overrightarrow{\mathrm{V}}\left(\mathrm{S_t}\right) + \alpha\left(\mathrm{R_{t+1+\tau}} + \gamma \mathrm{V}\left(\mathrm{S_{t+1+\tau}}\right) - \mathrm{V}\left(\mathrm{S_t}\right)\right)$$

$$Q\left(s_t, a_t\right) = Q\left(s_t, a_t\right) + \alpha\left(r_{t+1+\tau} + \gamma Q\left(s_{t+1+\tau}, a_{t+1+\tau}\right) - Q\left(s_t, a_t\right)\right)$$