CS6700 : Reinforcement Learning
Written Assignment #1

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

Name: M.V.A.Suhas Kumar — **Roll number:** EE17B109

- 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:
First let our algorithm (Inspiration from the proof of Median elimination algorithm)
Algorithm: 3rd quantile elimination (ϵ, δ)
1.Set S = A
2.$\epsilon_1 = \epsilon * y$,$\delta_1$=$\delta/2$,l=1
3.Sample every arm a ∈ S for $\frac{2}{\epsilon_l^2}$ln(x/δ_l) times, and let $Q(a_l)$ be empirical value denote its empirical value
4. Find the 3rd quantile of $Q(a_l)$, denoted by m_l
5. $S_{l+1} = S_l | a : Q(a_l) > m_l$
6.If $\mod S_l = 1. Then\ output\ S_l,$
Else $\epsilon_{l+1} = z\epsilon_l$; $\delta_{l+1} = \delta_l/2$; l =l+1 ; Go to 3

We still need to figure out x,y,z values using following method

PAC bound: Here we will find x and relation between y and z by ensuring the PAC bounds

$$Pr[max_{j\in S_l} Q_l(a_j) \leq max_{i\in S_{l+1}} Q_l(a_i) + \epsilon_l] < 1 - \delta_l$$

The above implies we loose the best arm of lth round with δ_l probability. Suppose even if we loose every round with δ_l. We can see that sum of all probabilities even if it runs to infinity is δ.

$$\delta_1 + \delta_2 .... = \delta/2 + \delta/4 + \delta/8 .... < \delta$$

And similarly for ϵ's which yields $\epsilon_{l+1} = \epsilon_l(1-y)$ which implies z = 1-y
We will now calculate the probability of loosing the best arm of lth round in lth round and equate it to δ_l
For eliminating the best arm there are two possible events:
1. Badly estimating($\epsilon_l/2$ away from true mean) the best arm (E1)
2.Even though if it does estimated correctly other 3/4 th of the sub optimal arms of lth round beat the best arm(E2).
Consider Event 1

$$P(E1) = pr[Q^*(a_l) < q^*(a_l) - \epsilon/2]$$

The above can be estimated from the chermoff's inequality

$$Pr(E1) = \delta_l/x$$

Now we consider Event 2
Where Event 1 is not true. we calculate the probability that an arm which is not an ϵ_l optimal arm is empirically better than the best arm.

$$Pr[Q_l(j) > Q_l(a_l^*)|Q_l(a_l^*) > q_l(a_l)^* - \epsilon_l/2] \leq Pr[Q_l(j) > q_l(j) + \epsilon/2|Q_l(a_l^*) > q_l(a_l)^* - \epsilon_l/2]$$

From chermoffs identity

$$Pr[Q_l(j) > q_l(j) + \epsilon/2|Q_l(a_l^*) > q_l(a_l)^* - \epsilon_l/2] = \delta_l/x$$

where j is not ϵ_l optimum
Expectation of Number of arms beating the optimal arm is

$$E[No.of bad arms|Q_l(a_l^*) > q_l(a_l)^* - \epsilon_l/2] = n\delta_l/x$$

where n is no of arms in that round
Using markov Inequality

$$Pr[E2] = Pr[No.of bad arms \geq 3n/4|Q_l(a_l^*) > q_l(a_l)^* - \epsilon_l/2] = 4/3 * \delta_l/x$$

$$P(E1) + p(E2) = \delta_l$$

From above we get x = 7/3 which ensures the PAC bounds for the algorithm
Now consider Algorithm complexity

In lth round there are n_l and each arm is sampled $(2/\epsilon_l^2)$*ln$(7/3\delta_l)$

There are total of $log_{4/3}(n)$ rounds

$$Total\ complexity = \sum_{l=0}^{l=log_{4/3}(n)} \frac{2n_l ln(7/3\delta_l)}{\epsilon_l^2}$$

$$\sum_{l=0}^{l=log_{4/3}(n)} \frac{2n_l ln(7/3\delta_l)}{\epsilon_l^2} = \frac{2n}{y^2}(\sum_{l=0}^{l=log_{4/3}(n)} (\frac{3}{4(1-y)^2})^{l-1} * (ln(7/3\delta) + l\ ln(2)\ )$$

$$\sum_{l=0}^{l=log_{4/3}(n)} \frac{2n_l ln(7/3\delta_l)}{\epsilon_l^2} < \frac{2nln(1/\delta)}{\epsilon^2 y^2} \sum_{l=0}^{l=\infty} (\frac{3}{4(1-y)^2})^{l-1}(lC' + C)$$

Above will converge only if $y < 1 - \frac{\sqrt{3}}{2}$.
we could take one value may y = 1/8
For that y

$$\sum_{l=0}^{l=log_{4/3}(n)} \frac{2n_l ln(7/3\delta_l)}{\epsilon_l^2} \leq O(\frac{nln(1/\delta)}{\epsilon^2})$$

conclusion: we sample each arm in lth round $\frac{2}{\epsilon_l^2}$ln$(7/3\delta_l)$
and $\epsilon_1 = \epsilon/8$ and $\epsilon_{l+1} = 7\epsilon_l/8$

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 think we can design a regret minimising algorithm which achieve better bounds than UCB. Knowing the true expected payoff will help us eliminate arms in the process of UCB and which mean lesser the number of arms in the processimplies faster convergence and regret will be minimised.
Algorithm:
1. Firstly we find the difference between the maximum expected payoff and minimum expected payoff. Let it be ϵ (Here ϵ = 4.6-0.9)
2.Run the UCB algorithm until all the arms estimated means are in the range of $\epsilon/2$

on each side.
3.Then eliminate the arm with least estimated value as it can't be the best arm.
4.Repeat again from step 1 without the eliminated arms.
5.Run the algorithm until one arm remains and from then on play greedy.

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: If we don't know which case we are going to get at each time step the best we could do is that we could pull the arm randomly at each time step

$$E[R] = P(A) * E[A] + P(B) * E(B)$$

$$E[A] = 0.5 * 0.1 + 0.5 * 0.2 = 0.15$$
$$E[B] = 0.5 * 0.9 + 0.5 * 0.8 = 0.85$$
$$E[R] = 0.5 * 0.15 + 0.5 * 0.85 = 0.5$$

   (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: If we know which case we are going to get at each time step then we could learn the optimal arm for each case (arm 2 for case A and arm 1 for case B) and play the arm repeatedly at each time step.

$$E[R] = P(A) * R(arm2|A) + P(B) * R(arm1|B)$$
$$E[R] = 0.5 * 0.2 + 0.5 * 0.9$$
$$E[R] = 0.55$$

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: Many tic-tac-toe positions appear to be different but they are the same. In that case when the updating the state value for a position we should also update all symmetric position state values. Here we are updating the values of some states even before we encounter them which would make us to expect faster convergence.

(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: "No", Suppose consider two symmetric positions,opponent is not taking advantage of symmetries means opponent is playing is better in one out of the two symmetric positions. For us to do better we should drive our algorithm to go to the symmetric position in which opponent is playing bad which would help us get more win. Preferring one symmetric position over others implies symmetrically equivalent positions necessarily need not have the same value.

(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: While playing with a normal opponent RL algorithm tries to learn to beat a specific opponent. While in self play where it plays against itself it tries to minimise self losses or it tries to reduce the opponent gains. Both the opponents in self play tries to exploit all the possible states and finally reaches a state where it can't loose. Which inturn I think is a very powerful strategy than what it learns against a specific opponent

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: In ego-centric learning agent distinguishes the surroundings based on what it see around itself. The advantages with this type of representation is that the state space would be small and would reduce a lot of computational cost. On the other hand ego-centric learning mainly focuses on immediate reward rather than long term reward. When the gamma value is increased to for see the future it might be non convergent systems when it leads to same state again and again.
For **example** when you go to certain state 1 in a grid by moving right yield to a state 2 where moving right gives reward. Getting reward from state 2 would boost the value to go right from state 1. when the map is large there a multiple locations in map where it is state(1') (Egocentric representation relatively it's same may not be absolutely same.Represented as 1' to ensure not to confuse from state 1 but they are the same relatively). Next time when you move to 1' you would take a action right(since you got reward by moving right from 1 last time) but now you wont get any reward, So this the value for taking action right from state 1. sometime u may get reward by moving right from state 1 and sometimes not. Like this it may keep on fluctuating and may not converge finally

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 k added to them. Derive the new value function V^π_{new} in terms of V^π, c and γ.

Solution: We know that from the definition of value function at a state s following the policy π as

$$v^\pi(s) = E_\pi[G_t|S_t = s] = E_\pi[\sum_{K=0}^{\infty} \gamma^K R_{t+K+1}|S_t = s]$$

$$v^\pi_{new}(s) = E_\pi[G'_t|S_t = s] = E_\pi[\sum_{K=0}^{\infty} \gamma^K R'_{t+K+1}|S_t = s]$$

Here given that R'_{t+K+1} =R_{t+K+1} + k
Substituting in the above Eqn:

$$v^\pi_{new}(s) = E_\pi[\sum_{K=0}^{\infty} \gamma^K R_{t+k+1}|S_t = s] + E_\pi[\sum_{K=0}^{\infty} \gamma^K k|S_t = s]$$

$$v^\pi_{new}(s) = v^\pi(s) + k\sum_{K=0}^{\infty} \gamma^K$$

$$v^\pi_{new}(s) = v^\pi(s) + \frac{k}{1-\gamma}$$

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: In both stochastic grid world and deterministic grid world to have the same have the same trajectories. we have to ensure for a given policy πwe need to have $P(s'|s)$ is equal in both worlds

Deterministic world:

$$p(s'|s) = \begin{cases} 1 - \epsilon + \frac{\epsilon}{|\mathcal{A}|} & s' = (s, \pi_d(s)) \\ \frac{\epsilon}{|\mathcal{A}|} & s' \neq (s, \pi_d(s)) \end{cases}$$

Stochastic world: In stochastic once we follow the policy then there is randomness arising from transitions probabilities
Here

$$P(s'|s) = Tr(s'|s, \pi_d(s))$$

To follow same trajectory both need to have same p(s'—s).

To specify the stochastic grid world here since since the states,actions are same as deterministic grid world it is sufficient to mention the transition probabilities. From above we get

$$Tr(s'|s, \pi_d(s)) = \begin{cases} 1 - \epsilon + \frac{\epsilon}{|\mathcal{A}|} & s' = (s, \pi_d(s)) \\ \frac{\epsilon}{|\mathcal{A}|} & s' \neq (s, \pi_d(s)) \end{cases}$$

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

Solution:

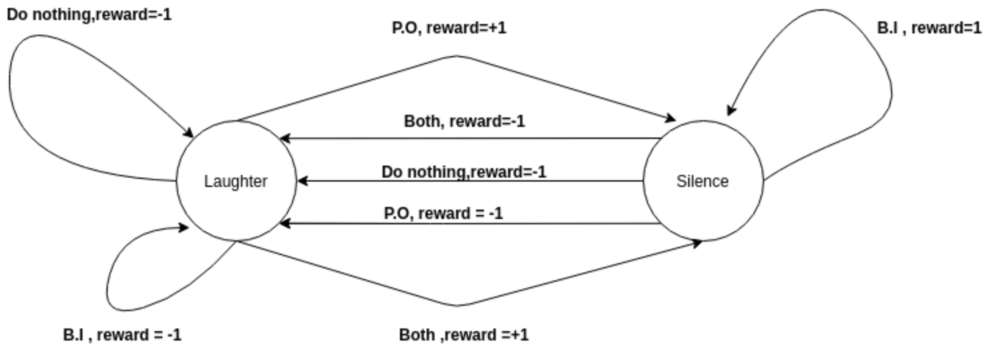
8. (7 marks) You receive the following letter:
Dear Friend, Some time ago, I bought this old house, but found it to be haunted by

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: Formulating above problem as MDP:
States : Laughter,Silence
Action sets : Do nothing , playing organ (P.O) , Burning Incense(B.I), Both (P.O $\cap$ B.I)
Rewards : +1 for transition into silent state , -1 for transition into laughter

State transitions:



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

Solution:
Initial policy π = B.I
Let random initial value function for states be : V(L)= -10 , V(S)= +10
$\gamma = 0.9$
Policy Iteration Round 1:
Step 1 : Policy Evaluation

$$V_\pi(L) = [-1 + 0.9 * -10] = -10$$

$$V_\pi(S) = [+1 + 0.9 * 10] = 10$$

Here for this initiation policy evaluation converge in 1 round. step 2: Policy Improvement:
Now for π^* we have four possibilities for next action (Do nothing (1)),(P.O(2)), (B.I(3)) ,(P.O$\cap$B.I(4)) and then following old policy B.I
For state L:

$$V_1(L) = -1 + 0.9 * -10 = -10$$

$$V_2(L) = +1 + 0.9 * 10 = 10$$

$$V_3(L) = -1 + 0.9 * -10 = -10$$

$$V_4(L) = +1 + 0.9 * 10 = 10$$

So we could pick either of 2 or 4 policy. Here I am picking 2.
New π(L) = P.O then B.I
For state S:

$$V_1(S) = -1 + 0.9 * -10 = -10$$

$$V_2(S) = -1 + 0.9 * -10 = -10$$

$$V_3(S) = 1 + 0.9 * 10 = 10$$

$$V_4(S) = -1 + 0.9 * -10 = -10$$

Here clearly policy 3 is better which means B.I so there no change in policy for silence state. New π^*(S) = π(S)
Policy Iteration Round 2:
$V(L) = -10$ $V(S) = 10$
π(L) = P.O then B.I , π(S) = B.I
step 1: Policy Evaluation

$$V_\pi(L) = [1 + 0.9 * 10] = 10$$

$$V_\pi(S) = [+1 + 0.9 * 10] = 10$$

Policy evaluation again converges in 1 round. step 2: policy Improvementt
For state L:
Now for π^* we have four possibilities for next action (Do nothing (1)),(P.O(2)), (B.I(3)) ,(P.O∩B.I(4)) and then following old policy

$$V_1(L) = -1 + 0.9 * 10 = 8$$
$$V_2(L) = +1 + 0.9 * 10 = 10$$
$$V_3(L) = -1 + 0.9 * 10 = 8$$
$$V_4(L) = +1 + 0.9 * 10 = 10$$

So we could pick either of 2 or 4. Picking policy 2.
which implies New π^*(L) = π(L)
so there no change in policy for Laughter state.
For state S:
Now for π^* we have four possibilities for next action (Do nothing (1)),(P.O(2)), (B.I(3)) ,(P.O∩B.I(4)) and then following old policy

$$V_1(S) = -1 + 0.9 * 10 = 8$$
$$V_2(S) = -1 + 0.9 * 10 = 8$$
$$V_3(S) = 1 + 0.9 * 10 = 10$$
$$V_4(S) = -1 + 0.9 * 10 = 8$$

We could see that 3rd policy is the best which is B.I . No improvement in policy
New π^*(S) = π(S)

In the second round of policy iteration there is no improvement in policy.
Final policy is:

$$\pi(L) = Playing\ organ\ and\ burning\ Incense\ forever$$
$$\pi(S) = Burning\ Incense\ forever$$

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

Solution: Play the organ when laughter is going on and then when it becomes silent burn incense forever.

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: Lets assume for simplicity our time step is 1 time unit and τ be a integer Which implies that corresponding action for the state at t^{th} time step will be applied at $t + \tau^{th}$ time step,which means that all the rewards obtained between t and $t + \tau$ will be independent of state at time step t. So we need to consider only rewards after $t + \tau$ for reward function

$$G_t = R_{t+\tau+1} + \gamma R_{t+\tau+2} + ... = \sum_{K=0}^{K=\infty} \gamma^K R_{t+\tau+K+1}$$

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

Solution: Hence from the above Reward notion TD(0) equation becomes:

$$V_\pi(s_t) = V_\pi(s_t) + \alpha[R_{t+\tau+1} + \gamma V_\pi(s_{t+\tau}) - V_\pi(s_t)]$$

References:
1. NPTEL video lectures by Ravindran sir at this URL
2.Median elimination algorithm paper at this URL1
3.Book: Richard S. Sutton and Andrew G. Barto. Introduction to Reinforcement Learning. MIT Press, Cambridge, MA, 2nd edition,2018.