CS6700 : Reinforcement Learning
Written Assignment #1

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

Name: H Vishal — **Roll number:** MM16B023

- 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. (4 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: PAC guarantee still holds in each of the rounds

$$P[\max_{j \epsilon S_l} p_j \leq \max_{i \epsilon S_{l+1}} p_i + \epsilon_l] \geq 1 - \delta$$

$$\implies P[\hat{p_1} \leq p_1 - \epsilon_1/2] + P[\hat{p_j} \geq p_i \mid \hat{p_1} \geq p_1 - \epsilon_1/2] \leq \delta_1$$

Let #bad be the number of arms which are not ϵ-optimal but are empirically better than the best arm.

$$[P[\#\text{bad} \geq \frac{3n}{4} \mid \hat{p_1} \geq p_1 - \epsilon_1/2] \leq \frac{fn\delta_1}{\frac{3n}{4}} = \frac{4x}{3}\delta_1,$$

$$\frac{4f\delta_1}{3} + f\delta_1 = \delta_1 \implies f = \frac{3}{7}$$

In order to find the relation for ϵ, consider $\epsilon_1 = \epsilon/k$, $\epsilon_l = \epsilon_1(1 - \frac{1}{k})$

$$\sum_{i=l}^{\infty} \epsilon_l = \epsilon, \implies \text{any finite sum } \sum_{i=l}^{M} \epsilon_l \leq \epsilon$$

Number of times each arm must be sampled in an iteration is given by

$$l = \frac{1}{(\frac{\epsilon_l}{2})^2} \log_2(\frac{7}{3\delta_l})$$

Solution:

$$\implies \sum_{l=1}^{\log_{4/3}(n)} \frac{n_l \log(\frac{7}{3\delta_l})}{(\frac{\epsilon_l}{2})^2} = 4 \sum_{l=1}^{\log_{4/3}(n)} \frac{3^{l-1}\frac{n}{4^{l-1}} \log_2(\frac{2^l 7}{3\delta})}{((1-\frac{1}{k})^{l-1}\frac{\epsilon}{k})^2}$$

$$\implies \frac{4}{k^2} \sum_{l=1}^{\log_{4/3}(n)} n(\frac{3k^2}{4(k-1)^2})^{l-1}(\frac{\log(\frac{1}{\delta})}{\epsilon^2} + \frac{\log(\frac{7}{3})}{\epsilon^2} + \frac{l\log(2)}{\epsilon^2})$$

For the series to converge, $\frac{3k^2}{4(k-1)^2} \leq 1$, $k = 8$ would satisfy this

$$\leq 256 \sum_{l=1}^{\log_{4/3}(n)} n(\frac{48}{49})^{l-1}(lC' + C) = O\Big(\frac{n\ ln(\frac{1}{\delta})}{\epsilon^2}\Big)$$

$\implies$ Sample complexity is same as of the original MEA algorithm

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: The difference between the expected payoffs are quite high. This provides the need for a better algorithm. Since we know the expectation value of the best arm, we define the quantity $\Delta_i = q_*(a_*) - q_*(a_i)$

UCB-improved: Maximize $Q(j) + \Delta_j\sqrt{\frac{2\ log\ t}{n_j}}$

Suppose for an arm j, $Q(j)$ - $\sqrt{\frac{2\ log\ t}{n_j}} < 3.1$, it is for sure that the lower bound is greater than 3.1. We can then keep eliminating the arms for which $Q(j) + \sqrt{\frac{2\ log\ t}{n_j}} <$ 4.6 until left with an optimal arm. Once the best possible action is determined in least possible action time steps, we can greedily minimize the regret and achieve better bounds as compared to the UCB algorithm.

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:** Case A: $q_*(1) = 0.1,\ q_*(2) = 0.2$, Case B: $q_*(1) = 0.9,\ q_*(2) = 0.8$, where 1 and 2 denote the respective actions
>
> Assumption: Face case A and case B with probability 0.5 each (uniform)
> $E[1] = 0.5(0.1) + 0.5(0.9) = 0.5,\ E[2] = 0.5(0.2) + 0.5(0.8) = 0.5$
>
> Since there is lack of information, the best we can do is to behave randomly and obtain an expected average reward $(R_1) = 0.5(0.5) + 0.5(0.5) = 0.5$.

   (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 what case is in front of us, the best way to go about would be to pick action 2 in times of case A and action 1 in times of case B
>
> By pulling the optimal arm every time (after learning since we don't know the true action values), the average reward obtained would be
> $R_2 = 0.5(0.2) + 0.5(0.9) = 0.55$
>
> Predictably, that the reward obtained is higher is in this scenario.

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:** In the case of tic-tac-toe, we can use a 4-fold symmetry which would essential reduce the board size to one-quarter of its original size and then define the possible set of actions on this new state space.

Using symmetry shrinks the state and the action space to have a more compact representation which will have better bounds and low computational cost.

(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 we are to enforce symmetry when the opponent is not taking advantage of it, we might lose handle on exploring biases of the opponent such as the opponent repeatedly making a bad move at a corner. This would also mean symmetrically equivalent positions don't necessarily have the same value 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: If the agent is playing against itself, it'd try to learn an optimal policy by having a mixture of both good (high reward) and bad (low or -ve reward) moves rather than the min max approach as it'd want both the agents to win. The policy might continue to adapt and converge at some point or keep fluctuating back and forth. So, convergence as such is not guaranteed.

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: Ego-centric learning can be adopted when the dynamics of a state is controlled only by the immediate neighbours of the environment (similar to cellular automata). The advantage of this is ofcourse, less storage requirement as there is no need to define functions to map far apart entities as they'd have have uncorrelated and independent dynamics.

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:

$$V_{new} = E_{\pi}\Big[\sum_{p=0}^{\infty} \gamma^p (R_t + k) \mid S_t = S\Big]$$

$$\implies V^{\pi}_{new} = V^{\pi} + E_{\pi}\Big[\sum_{p=0}^{\infty} \gamma^p * k \mid S_t = S\Big] = V^{\pi} + k\sum_{p=0}^{\infty} \gamma^p = V^{\pi} + \frac{k}{1-\gamma}$$

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: Stochastic grid world - Policy π_1 is deterministic. The agent takes a specific action as determined by π_1, but the environment induces stochasticity with probability $Pr(s'|s,a)$. But, $Pr(s'|s,a) = Pr(s',s)$ since the policy is deterministic.

Deterministic grid world - Even though the agent can take a policy $\pi_2(a|s)$, the final state is determined once the policy is fixed, i.e. $Pr(s'|s,a) = \pi_2(a|s)$

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

Solution: Yes, SARSA will converge to the same policy. Update rule for SARSA is,

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha * (R_{t+1} + \gamma * Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)).$$

The trajectory and transition probability is same in both worlds $\implies Q(S_t, A_t)$ is same (from Bellman Equation).

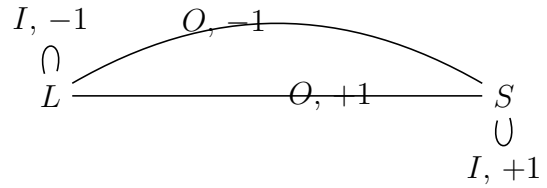
8. (7 marks) You receive the following letter:
Dear Friend, Some time ago, I bought this old house, but found it to be haunted by ghostly sardonic laughter. As a result it is hardly habitable. There is hope, however, for by actual testing I have found that this haunting is subject to certain laws, obscure 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-space S = {Laugher: L, Silent: S}
Actions = {Playing Organ: O, Burning Incense : I}
Discount factor $\gamma = 0.9$.

Assumptions: For the purpose of simplying the problem, let us assume that organ is not played if incense is burned and incense is not burned if organ is played. This would speed up the iterative process without compromising on the task of arriving at the optimal policy.

With the reward formalism mentioned in the question, we get the following state transition diagram.



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

Solution:

- Let us assume an initial estimate of policy, $\pi_0 = \{L : I,\ S : I\}$. In vector notation, the order is: L, S (for states).
- $p_{\pi_0} = \begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix}$. $r_{\pi_0} = \begin{matrix} -1 \\ +1 \end{matrix}$ $V_{\pi_0} = (I - \gamma p_{\pi_0})^{-1} r_{\pi_0} = \begin{matrix} -10 \\ 10 \end{matrix}$
- Now, $\pi_1(L) = argmax_a\{O :\ 1{+}0.9(10), I : -1{+}0.9(-10)\} = argmax_a\{O : 10, I : -10\} = O$
  $\pi_1(S) = argmax_a\{O :\ -1 + 0.9(-10), I :\ 1 + 0.9(10)\} = argmax_a\{O : -10, I : 10\} = I$
  Therefore, $\pi_1 = \{L : O,\ S : I\}$.
- $p_{\pi_1} = \begin{matrix} 0 & 1 \\ 0 & 1 \end{matrix}$. $r_{\pi_1} = \begin{matrix} +1 \\ +1 \end{matrix}$ $V_{\pi_1} = (I - \gamma p_{\pi_1})^{-1} r_{\pi_1} = \begin{matrix} 10 \\ 10 \end{matrix}$
- Now, $\pi_2(L) = argmax_a\{O :\ 1{+}0.9(10), I : -1{+}0.9(10)\} = argmax_a\{O : 10, I : 8\} = O$
  $\pi_2(S) = argmax_a\{O :\ -1{+}0.9(10), I : 1{+}0.9(10)\} = argmax_a\{O : 8, I : 10\} = I$
  Therefore, $\pi_2 = \{L : O,\ S : I\}$.
- $\pi_1 = \pi_2$. Thereforce, policy iteration converged to the optimal policy $\pi^* = \{L : O,\ S : I\}$.
- And optimal value function $V^* = V_{\pi_1} = \{L : 10, S : 10\}$.

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

Solution: Follow the optimal policy established to keep the room quiet. That is, play the organ if the room has laughter and if the room is quiet, burn the incense and stay away from the organ.

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: State is observed at time t and action is applied at time t+τ. Return for such as system is formulated as follows:

$$G_t = R_{t+\tau+1} + \gamma R_{t+\tau+2} + \gamma^2 R_{t+\tau+3} + ... = \sum_{k=1}^{\infty} \gamma^k R_{t+\tau+k}$$

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

Solution: The action value needs to be updated for $\hat{a}_t = a_{t-\tau}$ rather than considering Q for state s_t with value a_t. Thus, the TD(0) backup equation is,

$$V(S_t) \leftarrow V(S_t) + \alpha\big[R_{t+\tau+1} + \gamma V(S_{t+\tau+1}) - V(S_t)\big]$$