CS6700 : Reinforcement Learning
Written Assignment #1

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

Name: Neil Ghosh — **Roll number:** ME17B060

- 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:

n - Total no.of arms, n_l - No. of arms in the l^{th} round,
(ϵ_l, δ_l) - Values of ϵ and δ in the l^{th} round.

Let the values of ϵ_l and δ_l be the following -

$$\epsilon_1 = \epsilon/k,\ \delta_1 = \delta/2,\ n_1 = n$$

$$\epsilon_l = (\tfrac{k-1}{k})\epsilon_{l-1},\ \delta_l = \delta_{l-1}/2,\ n_l = 3n_{l-1}/4$$

$$\implies \epsilon_l = (\tfrac{k-1}{k})^{l-1}\epsilon/k,\ \delta_l = \delta/2^l,\ n_l = (\tfrac{3}{4})^{l-1}n$$

The values are chosen such that $\sum_{l=1}^{\infty}\delta_l = \delta$ and $\sum_{l=1}^{\infty}\epsilon_l = \epsilon$

No. of rounds $= \log_{\frac{4}{3}}(n)$, No. of samples in the l^{th} round $= \frac{2n_l}{\epsilon_l^2}\ln\frac{7}{3\delta_l}$

$$\implies \text{Total no. of samples} = \sum_{l=1}^{\log_{\frac{4}{3}}(n)}\frac{2n_l}{\epsilon_l^2}\ln\frac{7}{3\delta_l}$$

Substituting the values of n_l, δ_l and ϵ_l -

$$\implies 2\sum_{l=1}^{\log_{\frac{4}{3}}(n)}(\tfrac{k^2}{(k-1)^2})^{l-1}\tfrac{k^2}{\epsilon^2}(\tfrac{3}{4})^{l-1}n(\ln\tfrac{7.2^l}{3\delta})$$

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

For this sum to converge, $(\frac{3k^2}{4(k-1)^2})^{l-1}$ should be less than 1. The smallest integral value of satisfy this is k=8.

$$\implies 2\sum_{l=1}^{\log_{\frac{4}{3}}(n)}(\frac{48}{49})^{l-1}\frac{64n}{\epsilon^2}(\ln\frac{7}{3}+l\ln 2+\ln\frac{1}{\delta})$$

$$\leq \frac{128n\ln\frac{1}{\delta}}{\epsilon^2}\sum_{l=1}^{\infty}(\frac{48}{49})^{l-1}[C+lC'] \leq O(\frac{n\ln\frac{1}{\delta}}{\epsilon^2})$$

Therefore, even if we eliminate one-fourth of the arms instead of half, the sample complexity of the median elimination algorithm remains the same which is $O(\frac{n\ln\frac{1}{\delta}}{\epsilon^2})$.

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:

μ^* - True mean of the optimal arm, μ_i - True mean of the i^{th} arm,
$\hat{\mu_{i,t}}$ - Estimate of the mean of the i^{th} arm at time t.
UCB picks the action - $argmax_a[\hat{\mu_{a,t}}+\sqrt{\frac{2\ln t}{N_t(a)}}]$

To achieve better bounds, we basically need to ensure that the upper confidence bound of all the sub-optimal arms are lower than μ^* because then those arms will be picked with very low probabilty as there is a very high probabilty that the upper confidence bound of the optimal arm is going to be greater than μ^*.

$$\implies \hat{\mu_{a,t}}+\sqrt{\frac{2\ln t}{N_t(a)}} \leq \mu^* \text{ and } \hat{\mu_{a,t}} \leq \mu_a+\sqrt{\frac{2\ln t}{N_t(a)}} \text{ (with high probability)}$$

$$\implies \sqrt{\frac{2\ln t}{N_t(a)}} \leq \frac{\mu^*-\mu_a}{2}$$

The improved UCB should pick the action - $argmax_a[\hat{\mu_{a,t}}+\sqrt{\frac{2\ln t}{N_t(a)}}\Delta_a]$
where, $\Delta_a = \mu^* - \mu_i$

Since, μ^*=4.6 and the μ_i of the next best arm is 3.1, we only need to sample long enough so that the uncertainity term falls below $\frac{4.6-3.1}{2}=0.75$. After this we can keep on picking the greedy action to get higher rewards.

Since, all the true means are known, we can initialise the estimates of all the arms as the mean of all the true means instead of initialising them as 0. Using this the algorithm can find out the optimal arm sooner and minimise the regret.

$$\implies \frac{3.1+2.3+4.6+1.2+0.9}{5} = 2.42 \ \ \therefore \hat{\mu_{i,0}} = 2.42 \ \forall \ i$$

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 are not able to tell the case that we are facing, then we should randomly pick any arm. Expected reward of picking the first arm and the second arm are both the same.

   Expected Reward $= 0.5 \times 0.2 + 0.5 \times 0.8 = 0.5$ or $0.5 \times 0.1 + 0.5 \times 0.9 = 0.5$

   (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 the case that we are facing, then always pick the second arm in case A and pick the first arm in case B.

   Expected Reward $= 0.5 \times 0.2 + 0.5 \times 0.9 = 0.55$

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:

The tic-tac-toe board is symmetric both about the horizontal and vertical axis. So, for any current position on the board, there are 3 more positions which are symmetrical. By using symmetry we only have to store one-fourth of the states that were being stored, this will require less memory and decrease the time for converging to the optimal policy.

(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 is not taking advantage of symmetries that means that it may perform different actions in symmetrical states. It may perform poorly in one of the symmetrical states but not in others. We will not be able to exploit the opponents mistake if we use symmetry in this case so, we must also not use symmetry. It is not necessary that symmetrically equivalent positions must have the same value as the opponent may be performing differently in them even though they are symmetrical.

(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 instead of playing against a random opponent, the agent played against itself, the current state of the board will be seen differently by the two agents. A win for player 1 is a loss for player 2 and the value functions are updated accordingly. As more games are played the agent eventually learns the optimal value function and the optimal policy. After the optimal policy is found out, the players keep on playing the same set of moves again and again as there is no new data which is coming in for updating the value function and the policy. All the games played would result in a draw as no player would want to lose. So, a different policy is learnt as compared to the case where the agent plays against a random 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:

If we are using an ego-centric representation, the agent will try to maximise its immediate rewards more quickly. If we do not use an ego-centric representation, the agent will develop value functions which are better for long-term goals. Ideally, we would want some type of combination of both.

The ego-centric learner will try to avoid negative rewards in its immediate surroundings. This would mean that it would find the goal fewer number of times. A non ego-centric learner will reach the goal more number of times but it will also accumulate highly negative rewards along the way which are present in its surroundings.

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:

$$v_\pi(s) = E_\pi[G_t \mid S_t = s]$$
$$\implies v_\pi(s) = E_\pi[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... \mid S_t = s]$$

On adding a constant c to every reward, we will get a new value function V^{new}_π

$$v^{new}_\pi(s) = E_\pi[G'_t \mid S_t = s] = E_\pi[R'_{t+1} + \gamma R'_{t+2} + \gamma^2 R'_{t+3} + ... \mid S_t = s]$$

Since $R'_{t+k} = R_{t+k} + c$,

$$v^{new}_\pi(s) = E_\pi[R_{t+1} + c + \gamma(R_{t+2} + c) + \gamma^2(R_{t+3} + c) + ... \mid S_t = s]$$
$$\implies v^{new}_\pi(s) = v_\pi(s) + c(1 + \gamma + \gamma^2 + ...)$$
$$\therefore v^{new}_\pi(s) = v_\pi(s) + \tfrac{c}{1-\gamma} \implies V^\pi_{new} = V^* + \tfrac{c}{1-\gamma}$$

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:

The trajectory followed for both the cases is -
$S_t,\ A_t,\ R_{t+1},\ S_{t+1},\ A_{t+1},\ ...$

Let S_t = s and $\{a_1, a_2, ..., a_k\}$ are the actions available at state s

In the Deterministic Gridworld with Stochastic Policy -

$\pi(s) = argmax_a$ q(s,a) (with probability $1 - \epsilon$)
and $\pi(s) = \{a_1, a_2, ..., a_k\}$ (randomly picked action with probability $\frac{\epsilon}{n}$)
If $A_t = a,\ S_{t+1}$ becomes fixed. P(s,a,s') = 1 and S_{t+1} = s'.

In the Stochastic Gridworld with Deterministic Policy -

$\pi(s) = argmax_a$ q(s,a) (Given the state, the action becomes fixed)
If $A_t = a$ and $\{s_1, s_2, ..., s_k\}$ are the states that we can transition to,
P(s,a,s') = $1 - \epsilon$ (for a state s') and
P(s,a,s') = $\frac{\epsilon}{n}$ (for each of the remaining states in $\{s_1, s_2, ..., s_k\}$)

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

Solution:

The SARSA update equation is -
$q_{k+1}(s, a) \leftarrow q_k(s, a) + \alpha[r + \gamma q_k(s', a') - q_k(s, a)]$

If S_t = s, the probability of S_{t+1} being s' (i.e. P(s' | s)) is the same for both the worlds. P(s' | s) = $\pi(a|s) \cdot P(s, a, s')$

Since, the same state is going to be visited with equal probability for both the worlds, eventually the value functions of both the worlds for each of the same states will be equal. Therefore, SARSA will converge to the same policy for both the worlds.

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:

Set of States $\mathcal{S}$ = {Laughing (L), Silent (S)}

Set of Actions $\mathcal{A}$ = {Only Organ (O,I'), Only Incense (O',I), Both Organ and Incense (O,I), Neither Organ or Incense (O',I')}

State Transitions (s,a,s') = {(L,(O,I),S), (L,(O,I'),S), (L,(O',I'),L), (L,(O',I),L), (S,(O,I),L), (S,(O,I'),L), (S,(O',I'),L), (S,(O',I),S)}

R(L,(O,I),S) = 1, R(L,(O,I'),S) = 1, R(L,(O',I'),L) = -1, R(L,(O',I),L) = -1, R(S,(O,I),L) = -1, R(S,(O,I'),L) = -1, R(S,(O',I'),L) = -1, R(S,(O',I),S) = 1

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

Solution:

$\pi_0(L) = (O', I)$, $\pi_0(S) = (O', I)$ and $v_0(L) = 0$, $v_0(S) = 0$, $\gamma = 0.9$
$v_0(s) = \sum_a \pi_0(a|s) \sum_{s'} P(s, a, s')[R(s, a, s') + \gamma v_0(s')]$ (Policy Evaluation)

$$v_0(L) = 1 \cdot 1 \cdot [-1 + \gamma(0)] = -1,\ v_0(S) = 1 \cdot 1 \cdot [+1 + \gamma(0)] = +1$$
$$v_0(L) = 1 \cdot 1 \cdot [-1 + \gamma(-1)] = -(1 + \gamma),\ v_0(S) = 1 \cdot 1 \cdot [+1 + \gamma(1)] = (1 + \gamma)$$
$$v_0(L) = 1 \cdot 1 \cdot [-1 + \gamma(-1(1 + \gamma))] = -(1 + \gamma + \gamma^2),$$
$$v_0(S) = 1 \cdot 1 \cdot [+1 + \gamma(1 + \gamma)] = (1 + \gamma + \gamma^2)$$
$$\implies v_{\pi_0}(L) = -(1 + \gamma + \gamma^2 + \gamma^3 + ...) = -\tfrac{1}{1-\gamma} = -\tfrac{1}{1-0.9} = -10$$

$$\implies v_{\pi_0}(S) = 1 + \gamma + \gamma^2 + \gamma^3 + ... = \frac{1}{1-\gamma} = \frac{1}{1-0.9} = 10$$

$$\pi_1(s) = argmax_a(P(s,a,s')[R(s,a,s') + \gamma v_0(s')]) \text{ (Policy Improvement)}$$

$$\pi_1(L) = argmax_a\{(O,I) : 1 \cdot [1 + \gamma(10)] = 10,\ (O',I) : 1 \cdot [-1 + \gamma(-10)] = -10,\ (O,I') : 1 \cdot [1 + \gamma(10)] = 10,\ (O',I') : 1 \cdot [-1 + \gamma(-10)] = -10\} = (O,I)\ or\ (O,I')$$

$$\pi_1(S) = argmax_a\{(O,I) : 1 \cdot [-1 + \gamma(-10)] = -10,\ (O',I) : 1 \cdot [1 + \gamma(10)] = 10,\ (O,I') : 1 \cdot [-1 + \gamma(-10)] = -10,\ (O',I') : 1 \cdot [-1 + \gamma(-10)] = -10\} = (O',I)$$

$$v_1(L) = 0.5 \cdot 1 \cdot [1 + \gamma(10)] + 0.5 \cdot 1 \cdot [1 + \gamma(10)] = 10,\ v_1(S) = 1 \cdot 1 \cdot [1 + \gamma(10)] = 10$$
$$\implies v_{\pi_1}(L) = 10,\ v_{\pi_1}(S) = 10$$

$$\pi_2(L) = argmax_a\{(O,I) : 1 \cdot [1 + \gamma(10)] = 10,\ (O',I) : 1 \cdot [-1 + \gamma(10)] = 8,\ (O,I') : 1 \cdot [1 + \gamma(10)] = 10,\ (O',I') : 1 \cdot [-1 + \gamma(10)] = 8\} = (O,I)\ or\ (O,I')$$

$$\pi_2(S) = argmax_a\{(O,I) : 1 \cdot [-1 + \gamma(10)] = 8,\ (O',I) : 1 \cdot [1 + \gamma(10)] = 10,\ (O,I') : 1 \cdot [-1 + \gamma(10)] = 8,\ (O',I') : 1 \cdot [-1 + \gamma(10)] = 8\} = (O',I)$$

$\pi_1 = \pi_2$ implies that both are optimal policies. $\therefore \pi_1 = \pi_2 = \pi^*$

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

Solution:

The optimal policy should be followed to keep the house quiet (as found in part (b)). If the current state is laughter, play the organ (with or without lighting the incense) and then keep lighting the incense indefinitely without playing the organ. If the current state is silent, keep lighting the incense indefinitely without playing 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:

The trajectory followed will be of the form -

$$S_t,\ A_t,\ R_{t+1},\ S_{t+1},\ A_{t+1},\ R_{t+2},\ S_{t+2}\ ...\ S_{t+\tau},\ A_{t+\tau},\ R_{t+\tau+1},\ S_{t+\tau+1}\ ...$$

where, $A_{t+\tau} = \pi(S_t),\ A_{t+\tau+1} = \pi(S_{t+1})$ and so on ...

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

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

Solution:

$R_{t+\tau+1} + \gamma V_k(S_{t+\tau+1})$ will be the value of the new sample.

The TD(0) backup equation will be -

$V_{k+1}(S_t) \leftarrow V_k(S_t) + \alpha[R_{t+\tau+1} + \gamma V_k(S_{t+\tau+1}) - V_k(S_t)]$