CS6700 : Reinforcement Learning
Written Assignment #1

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

Name: Aniruddha Roy — **Roll number:** EE18D031

- 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: In this new modification of the algorithm , we will eliminate $\frac{1}{4}$ of the arms after each round.
Now consider the the l th round-

$$A_l : \{\max_{i \in S_{l+1}} q^*(i) \leq \max_{j \in S_l} q^*(j) - \epsilon_l\}$$

$$A_1 : \{Q_l(a_l^*) < q^*(a_l^*) - \frac{\epsilon_l}{2}\}$$

Using total probability law,

$$\Pr(A_l) \leq \Pr(A_1) + \Pr(A_l | A_l^c) \tag{1}$$

Using Chernoff-Hoeffding bound,

$$\Pr(A_l) \leq e^{-\frac{n_l \epsilon_l^2}{2}} \tag{2}$$

$$A_2 : \{Q_l(a) \geq Q_l(a_l^* | A_1^c\} \cap q^*(a) \leq q^*(a_l^*) - \epsilon_l \tag{3}$$

$$\Pr(A_2) \leq e^{-\frac{n_l \epsilon_l^2}{2}} \tag{4}$$

$$A[n_b] \leq (|S_l - 1|)e^{-\frac{n_l \epsilon_l^2}{2}} \leq |S_l|e^{-\frac{n_l \epsilon_l^2}{2}} \tag{5}$$

All arms enters into the next round that is $(l+1)$ th, then the number of bad arms $n_b = 1 - \frac{|S_l|}{4} = 3\frac{|S_l|}{4}$

Using Markov inequality

$$\Pr(A_l|A_1^c) \leq \frac{|S_l|e^{-\frac{n_l \epsilon_l^2}{2}}}{3\frac{|S_l|}{4}} = 4\frac{|S_l|e^{-\frac{n_l \epsilon_l^2}{2}}}{3|S_l|} \tag{6}$$

Using equation (1)

$$\Pr(A_l) \leq \frac{7e^{\frac{n_l \epsilon_l^2}{2}}}{3} \tag{7}$$

Now, to bound the (7) by δ_l , we will get the sample size

$$n_l = 2\frac{1}{\epsilon_l^2} log(\frac{7}{3\delta_l}) \tag{8}$$

So, from (8) it is clear that sample size is greater than with comparison to MEA algorithm. Now, we prove (ϵ, δ) PAC bound for this algorithm. The total sum value of ϵ_l will be consists with in the ϵ . Here, the initialization is

$$\epsilon_1 = \frac{\epsilon_l}{4}$$
$$\epsilon_{l+1} = 1 - \frac{\epsilon_l}{4} = \frac{3\epsilon_l}{4}$$

We have to choose the δ_l value $\frac{1}{2^l}$, so that total sum of δ_l will be lie with in δ

We can verify that sample complexity is still, $O(\frac{k}{\epsilon_l^2} log(\frac{1}{\delta}))$

$$\sum_{l=1}^{m} n_l = O(\frac{k}{\epsilon_l^2} log(\frac{1}{\delta})) \tag{9}$$

where, $m = log_{\frac{4}{3}}(k)$. This algorithm still in (ϵ, δ) PAC.

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: In MAB problems model the exploration and exploitation dilemma. Upper confidence bound algorithm is one of the methods to find the optimal strategy. We can minimize the regret using Thompson sampling (TS)-based method. Thompson proposed a randomized method to minimize regret. The main idea this, we have to assume a prior distribution on the parameters of reward estimation of every arm. Then select the best arm according to its posterior probability. Its is well-developed theory and proved that the select arm is close to optimal. Using TS, it is possible to minimize the regret so that we can achieve better regret bounds than USB. This algorithm is based on a randomized Bayesian method. One of the recent works [1] which theoretically shows that we can achieve logarithmic expected regret for the MAB problems with the comparison with UCB.

Theorem 1 *For the two armed stochastic bandit problem (K =2) , TS algorithm has expected regret*
$\mathbb{E}[R(T)] = O(\frac{lnT}{\Delta} + \frac{1}{\Delta^3})$ *in time* T*, where* $\Delta = \mu_1 - \mu_2$

Theorem 2 *For the* K *armed stochastic bandit problem, TS algorithm has expected regret*
$\mathbb{E}[R(T)] \leq O((\sum_{a=2}^{K} \frac{1}{\Delta_a^2})^2 lnT)$ *in time* T*,where* $\Delta_i = \mu_1 - \mu_i$

The above two theorems are proved by [1]. They have shown that, we can bound the expected regret of TS. They assumed that the first arm is the unique optimal arm, i.e., $\mu^* = \mu_1 > argmax_{i \neq 1}(\mu_i)$. Without loss of generality , since adding more arms with $\mu_i = \mu^*$ can decrease the expected regret[1]. Notice that, here bound are optimal but depends on Δ_i and the constant factor in $O(h)$.
We can modify the UCB algorithm to achieve the better bounds. The authors [2,3] are modified the confidence interval part of UCB to $\sqrt{\frac{2ln(t\Delta_i^2)}{n_i}}$. The Δ_i is measure of distance between a sub optimal and optimal arm i. Here, Δ_i is the unknown for the learner.

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: Let denote the arms are a_1 and a_2.
The expected reward for action-1 across both the cases
$\mathbb{E}[a_1] = 0.1 \times 0.5 + 0.9 \times 0.5 = 0.50$
The expected reward for action-2 across both the cases
$\mathbb{E}[a_2] = 0.2 \times 0.5 + 0.8 \times 0.5 = 0.50$
Since expected reward for both the arms are equal are equal. That's why choice of arm will not be matter in the long run for both the cases.

(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: This is an associative search task. Here the task is non-stationary so the agent will try to track the best action because it changes with time. We can consider case-A is task-a and case-B is task-B. In this situation, the main objective is to learn a policy. A policy is basically maps from a given state or situations to the actions that are optimal for those situations. This method is trial-and-error learning to searching for optimal actions. So, this process involves learning a policy.
Here, the policy is deterministic. Lets, the situations are denoted by s_1 and s_2. The actions are a_1 and a_2 for both the tasks. Now we define a policy like this,

$$\pi(a_1|s_1) = 0$$
$$\pi(a_2|s_1) = 1$$
$$\pi(a_1|s_2) = 1$$
$$\pi(a_2|s_2) = 0$$

So, the policy is a probability distribution over the action in a given state. From the above policy it is agent senses the situation s_1, it will always pick the best action that is action a_2 and when senses the situation s_2, it will pick the best arm that is action a_1.

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 algorithm will be faster because it is possible to use four axes of symmetry. Every axis of symmetry contains four states. That means the number of states can be reduced using symmetry. So, this symmetry reduces the state-space size, which is an advantage of using this method. Hence, it can be easy to learn the optimal policy under this reduced state space.

(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 an opponent did not take advantage of symmetries, that means it could be possible the overall performance will be worse. Because, opponent of this game might be play different moves. That means our estimate of probable winning chance from a symmetrical point of view could be wrong. So, it is not true that symmetrically positions should necessarily 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: In this case, the reinforcement learning algorithms will learn self-play, that is how to play against itself. In most of the situations, algorithms will pick greedy moves. Because of the self-play scenario, the opponent player will do the same thing. This playing is like a min-max, and the algorithm will learn this way of playing this game.

Let us take an example to explain the above arguments. Suppose, we are in a present state (s) in a board, where the probability of winning chance is 0.6. Suppose there are three states s_1,s_2,s_3. Now, from the state s to we can reach one of the three states. We can move to one one of the sate where the probability of winning chance is more based on the estimate of our current state. For example, if we move to the state s_3 there is the probability of winning chance is 0.7. So, based on the current state, we already know where we are moving. Our opponent player uses the same learning algorithm, but he is one move behind. That means in this state opponent player minimizes the maximum winning chance for us. This is a min-max play. In the game-theoretical point of solution of this game called, the Nash equilibrium. Our opponent player tries to minimize the maximum payoff for us.. Hence reinforcement learning algorithm will learn to the min-max way of play.

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 representations are based on an agent's current position in the world. Agents are only worried about the position of the objects in the world, which is relative to the agent. That means, the agent learns the situations based on immediate surroundings. So, the agent gets the immediate benefit of rewards without no delay. That means, the agent learns fast and may not to take actions which may be reaching a state where a reward may be negative. In contrast, we can say, agents, learns the surrounding so quickly so that it can reach a state which gives positive rewards. So, based on immediate learning the environment quickly, the agent may reach a state where the chance of getting a positive reward may be very high. This is the advantage of this representations.

Here, agents try to reach a state based on immediate actions, which may give positive rewards but due to lack of exploration of other states based on immediate learning, the agent may not get the high positive reward in the long run. This is the disadvantage for the long run.

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: The value function of a state (s) under policy π, is denoted by $V^\pi(s)$. This value function $V^\pi(s)$ will give expected return, starting from state s and following a policy π. According to the definition, we can write,

$$\begin{aligned} V^\pi(s) &= \mathbb{E}_\pi[G_t|S_t = s] \\ &= \mathbb{E}_\pi[(R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ...)|S_t = s](\text{ where } 0 < \gamma < 1) \\ &= \mathbb{E}_\pi[\sum_{m=0}^{\infty} \gamma^m R_{t+m+1}|S_t = s] \end{aligned}$$

Now, we are adding constant k to all rewards. So, the new value function will be

$$\begin{aligned} V^\pi_{new}(s) &= \mathbb{E}_\pi[(k + R_{t+1}) + \gamma(k + R_{t+2}) + \gamma^2(k + R_{t+3}) + ...|S_t = s] \\ &= \mathbb{E}_\pi[\sum_{m=0}^{\infty} \gamma^m(R_{t+m+1} + k)|S_t = s] \end{aligned}$$

Apply linearity of expectation,

$$\begin{aligned} V^{\pi}_{new}(s) &= \mathbb{E}_{\pi}[\sum_{m=0}^{\infty} \gamma^m (R_{t+m+1}) | S_t = s] + \mathbb{E}_{\pi}[\sum_{m=0}^{\infty} \gamma^m k | S_t = s] \\ &= \mathbb{E}_{\pi}[\sum_{m=0}^{\infty} \gamma^m (R_{t+m+1}) | S_t = s] + k\mathbb{E}_{\pi}[\sum_{m=0}^{\infty} \gamma^m | S_t = s] \\ &= V^{\pi}(s) + k.\frac{1}{1-\gamma} \end{aligned}$$

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 this problem, ϵ soft policy is given. We can think about in that way ϵ greedy policy is a subset of a ϵ soft policy. This policy take every action with probability at least ϵ over cardinally of the action set. Over ϵ soft policy agent can continuously explore continuously. The ϵ soft policies are always stochastic deterministic policy specify a single action to take in each state stochastic policies instead specify the probability of taking action in each state in epsilon [4]. If our policy always gives at least ϵ probability to each of the action, it is not possible to converge to a deterministic best policy exploring at starts could be used to find the best policy. But, that is on soft policies can only be used to find the best ϵ soft policy. In general, this policy can not perform better than the optimal policy. However, it may be performs reasonably.

We have deterministic policy in the stochastic gridworld and ϵ soft policy in a deterministic world. We have given that both the world will give same trajectories $\forall a \in \mathcal{A}, \forall s \in \mathcal{S}$ .Suppose deterministic policy maps from the state(s_1) to the action a_1. Both world will produce same trajectory that means deterministic policy has to be equal to the transition probability of the stochastic gridworld. Let us consider two state s_1 and s_2 and that means $Pr(s_2|s_1, a) = \pi(a|s_1)$

So for the deterministic world ,

$$Pr(s_2|s_1, a) = \pi(a|s_1) \forall s_1, s_2 \in \mathcal{S}, \forall a \in \mathcal{A}$$

this is also for stochastic grid world. That is both are generating same trajectories. Otherwise, $Pr(s_2|s_1, a) = 0$. That means they are not generating same trajectories.

For on-policy control methods, probability of selecting non-greedy is $\frac{\epsilon}{|\mathcal{A}|}$. So, for greedy actions probability is $1 - \epsilon + \frac{\epsilon}{|\mathcal{A}|}$

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

Solution: The Q function for the two world may not same. So, SARSA on the two worlds might not be converge to the same policy. But, the expected SARSA might be 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: Here, the states are laughter (L)and quiet(Q). So, the state set (S) is $\{L, Q\}$.

Here, actions are- (i) play organ only (O), (ii) burn incense only (I) , (iii) do both (B) and (iv) do nothing(N).

So, here action set (A) = $\{O, I, B, N\}$

The reward will be $+1$ on any transition into the silent state and -1 on any transition into the laughing state. The state transition diagram is shown in the figure.1 (refer to the page-11)

The reward set is , $R = \{+1, -1\}$ and,the discount factor $\gamma = 0.9$

Here, MDP is $\langle S, A, P, R, \gamma \rangle$

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

Solution: Here,the policy given is always burning incense and not playing the organ. So, we can define the policy like-

$\pi(I|.) = 1$ and $\pi(O|.) = 0$, that means whatever is my current state, if I will take action I, the probability is always be 1 (because this is my best action from the action sets) and for burning incense it will be 0. We know the bellman equation-

$$V^{\pi}_{k+1}(s) = \sum_{a \in A} \pi(a|s).[r + \sum_{s' \in S} P(a|s, s')V^{\pi}_{k}(s')]$$

At $K = 0$:

$V^{\pi}(L) = V^{\pi}(Q) = 0$

At $K = 1$:

$V^{\pi}(L) = 1.[-1 + 0.9(1 \times 0 + 0 \times 0))] + 0.[...] + 0.[...] + 0.[...] = -1$

$V^{\pi}(Q) = 1.[+1 + 0.9(0 \times 0 + 1 \times 0)] = 1 + (0.9 \times 0) = 1$

At $k = 2$:

$V^{\pi}(L) = 1.[-1 + 0.9(1 \times -1 + 0 \times 1))] = -1.9$

$V^{\pi}(Q) = 1.[+1 + 0.9(0 \times -1 + 1 \times 1)] = 1 + (0.9 \times 0) = 1.9$

At $k = 3 : V^{\pi}(L) = -2.71$ and $V^{\pi}(Q) = +2.71$

At $k = 4 : V^{\pi}(L) = -3.433$ and $V^{\pi}(Q) = +3.433$

At $k = 5 : V^{\pi}(L) = -4.087$ and $V^{\pi}(Q) = +4.087$

At $k = 6 : V^{\pi}(L) = -4.681$ and $V^{\pi}(Q) = +4.681$

At $k = 7 : V^{\pi}(L) = -5.212$ and $V^{\pi}(Q) = +5.212$

At $k = 8 : V^{\pi}(L) = -5.691$ and $V^{\pi}(Q) = +5.691$

.....

At $k = \infty : V^{\pi}(L) = -10$ and $V^{\pi}(Q) = +10$

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

Solution: If there is laughter(L) the best action is play the organ only. But if the room is quiet (Q), do not play the organ and burn incense only.

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: In this problem, the agent takes an action on observing the state at time t. The action is applied after time delay τ . That is at time instant $t + \tau$ the action is applied to the system. So, this process is associated with time-delay. We know discounted return, if there is no delay with infinite time horizon

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ...$$

Now, with time-delay τ , it will be-

$$G_t = R_{(t+\tau)+1} + \gamma R_{(t+\tau)+2} + \gamma^2 R_{(t+\tau)+3} + ...$$

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

Solution: The update rule of TD method -

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

For TD update, target = $R_{t+1} + \gamma V(S_{t+1})$, this is TD(0). This is also called, one-step TD, special case of TD(λ). TD error measures the difference between the estimated value pf state and target value.

$$\delta_t = R_{t+1} + \gamma V(S_{t+1}) - V(S_t)$$

Now for time-delay case-

$$\delta_t = R_{t+\tau+1} + \gamma V(S_{t+\tau+1}) - V(S_t)$$

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

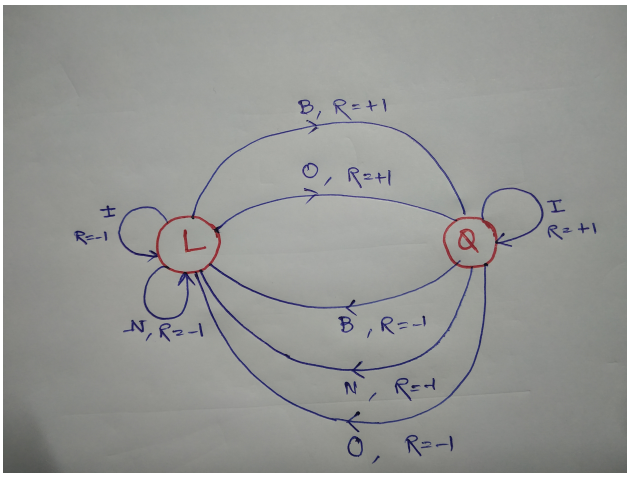

Figure 1: State diagram (question 9.a)

References

1. Shipra Agrawal, Navin Goyal, Analysis of Thompson Sampling for the multi-armed bandit problem, *JMLR: workshop and conference proceeding*, vol. 23, 2012

2. Rajeev Agarawal, Sample mean based index policies with $O(logn)$ regret for the multi-armed bandit problem, *Advances in Applied Probability*, vol.27 ,1995 , pp. 1054–1078.

3. Peter Auer and Ronald Ortner, UCB Revisited: Improved regret bounds for the stochastic multi-armed bandit problem, *Periodica Mathematica Hungarica*, vol. 61, 2010, pp. 55–65

4. Reinforcemet learning, Coursera Online course

5. Richard S. Sutton and Andrew G. Barto, Reinforcement learning : an introduction, 2nd edition, The MIT press.