CS6700 : Reinforcement Learning
Written Assignment #1

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

Name: Arabhi Subhash — **Roll number:**cs17b005

- 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: Removing less arms (here one-fourth) in every step improves the δ. In fact the $pr(bad >= 3|S_l|/4) <= 4S_l/9$ in the proof of algorithm will reduce the probability of bad events even less than $7\delta/9$.
Sample Complexity :

for lth round - no. of samples $= \frac{4 n_l \ln(3/\delta_l)}{\epsilon_l^2}$

1) $\delta_l = \delta/2^l$ 2) $\epsilon_l = (3/4)^{l-1}\epsilon/l$

3) $n_l = (3/4)^{l-1} n$

complenity $= \sum_{l=1}^{\log_{4/3} n} \frac{n_l \ln(3/\delta_l)}{(\epsilon_l/2)^2}$

$$= 64 \sum_{l=1}^{\log_{4/3} n} n\left(\frac{4}{3}\right)^{l-1}\left(\ln(1/\delta) + \ln 3 + l\ln 2\right) 1/\epsilon^2$$

$$\leq c\, \frac{n \ln(1/\delta)}{\epsilon^2} \sum_{l=1}^{\log_{4/3} n} (c_1 + l)(4/3)^{l-1} \quad \text{— ①}$$

Here we cant change the summation to ∞ as $4/3 > 1$ it blows up so consider

$$(4/3)^t \leq n < (4/3)^{t+1}, \quad t \in \mathbb{N} \cup \{0\}$$

$$\text{①} = c_1 \underbrace{\sum_{l=1}^{t} (4/3)^{l-1}}_{GP} + \underbrace{\sum_{l=1}^{t} (4/3)^{l-1} \times l}_{AGP}$$

$$= c_1\left[\frac{(4/3)^t - 1}{4/3 - 1}\right] + \left[\frac{t(4/3)^t}{4/3 - 1} + \frac{(4/3)^t - 1}{(4/3 - 1)^2}\right]$$

$$(4/3)^t < n, \quad t < \log n$$

$$\leq c_2 n + c_3 n\log n + c_4$$

① $= o(n\log n)$

so complexity $= n^2/\epsilon^2 \log(1/\delta) \log n$

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: We can do Thompson sampling algorithm to minimize regret and achieve better bounds. The reason is that in Thompson sampling we assume some distribution like Gaussian for each arm and here we can use our payoffs to improve this estimates.
One way of improving UCB is by assigning weights to each arm. The weights are probabilities of max payoffs (4.6 here) in Gaussians formed by sample mean and variance of each arm. These weights make algorithm converge faster towards max-payoff.

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: As we don't know the state, its the case of contextual bandits (associative search) here similar to a RL problem we can compare expected rewards and find the best arm.
   E(arm-1) = 0.5 * 0.1 + 0.5 * 0.9 = 0.5
   E(arm-2) = 0.5 * 0.2 + 0.5 * 0.8 = 0.5
   As both are giving same it doesn't matter which are we pull and the expected reward will also be **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: As we know which state each step is this contextual bandit problem turns into two separate deterministic-bandit problems.
   For case A : We always select arm 2 i.e. reward is always .2 and for case B : we select arm 1 with reward .9. Hence total expected reward is **0.5*(.9 + .2) = .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: By considering symmetry we can reduce the state-space i.e. symmetric states can be considered as equals. This will easy up our exploration work and algorithms will converge faster.

(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 opponent is not taking the advantage then there is a possibility that he might have a bias towards one among several symmetric states. If this happens and we are using symmetry then we aren't exploring that bias to our advantage. So no, you should not use it if opponent is not taking this advantage.

(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 one can use the advantage of symmetry as we know that opponent is unbiased towards a specific one among symmetric ones. As said in first question, this will reduce the state space and algorithms will converge faster. The resulting policy here and in the above B problem will be different as for the same fact there can be some bias among symmetric states unlike here.

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 egocentric learning we look at the world and space from us, states will be the positions you see or results of very next action. The other one is allocentric approach which maintains the map of whole world and states will be markings in a map. The advantage of this approach is that we can save space for maintaining states as they are with respect to your present position. Disadvantage is following egocentric approach we tend to look for immediate benefit rather than the total benefit in later steps.

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:

Definition of $V^\pi(s)$ discounted

$$V^\pi(s) = E\{G_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} \ldots + \gamma^{T-t} R_T\}$$

Here it is given that the horizon is ∞

$$V^\pi(s) = E\{R_t + \gamma R_{t+1} \cdots\}$$

$$V^\pi_{new}(s) = E\{R_{t_{new}} + \gamma R_{t+1_{new}} \cdots\}$$

$$= E\{k + R_t + k\gamma + \gamma R_{t+1} \cdots\} \text{ - given}$$

$$= k(1 + \gamma + \gamma^2 + \cdots) + E\{R_t + R_{t+1} \cdots\}$$

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

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:
Deterministic Gridworld :
Here one's actions directly result in his state change. Given that we are gonna follow deterministic policy (selecting $\pi_d(S)$) for $1 - \epsilon$ times and uniform policy rest of time. So here

P(S'/S) = 1 - ϵ + $\epsilon/|A|$ for a = $\pi_d(S)$
= $\epsilon/|A|$ for a $\neq \pi_d(S)$
Stochastic Gridworld :
Here a complete deterministic policy is applied but here action doesn't directly result in final state. There is another transition probability function (Tr) which determines the probability of going onto a state given present state and applied action. Given that both follow same trajectory hence the P(S'/S) and Tr(S'/S,$\pi_d(S)$) should be equal.
Tr(S'/S,$\pi_d(S)$) = 1 - ϵ + $\epsilon/|A|$, S' is result of action $\pi_d(S)$ in det. world
= $\epsilon/|A|$, S' is result of action $\neq \pi_d(S)$ in det. world
This transition function defines the stochastic grid world.

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

Solution: No, They will not converge to same action because we take other than best, the action performed currently to learn the Q-value. This change in update step will result in deviation of final convergence as transition probabilities will differ with action even for same transition among states in stochastic world.

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

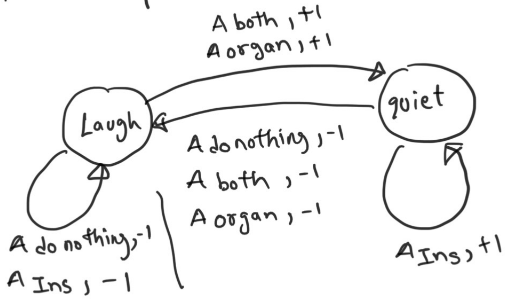
(a) (3 marks) Formulate this problem as an MDP (for the sake of uniformity, formulate it as a continuing discounted problem, with $\gamma = 0.9$. Let the reward be +1 on any transition into the silent state, and -1 on any transition into the laughing state.) Explicitly give the state set, action sets,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

Solution: Formulating above problem as MDP!

Actions
A do nothing
A play organ
A Insence
A both

States
S Laughter
S quiet

Rewards
+1 - getting into quiet
-1 - getting into laughter



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

Solution: Here N = Do Nothing, P = Play Organ, I = Burn Incense, B = Do Both.

Starting Policy

$\pi = \mathrm{I}$

$$V_\pi(L) = -1 + 0.9 * -1 + 0.9^2 * -1 - ... = -10$$

$$V_\pi(S) = 1 + 0.9 * 1 + 0.9^2 * 1 - ... = 10$$

Policy Improvement I

π_1 = N v I v P v B and follow the previous policy

$$V_{\pi_1}(L) = -1 + 0.9 * -10 = -10$$

$$V_{\pi_1}(S) = 1 + 0.9 * 10 = 10$$

Q-Update :

$$Q(L, N) = -1 + 0.9 * -10 = -10$$

$$Q(S, N) = -1 + 0.9 * -10 = -10$$

$$Q(L, P) = 1 + 0.9 * 10 = 10$$

$$Q(S, P) = -1 + 0.9 * -10 = -10$$

$$Q(L, I) = -1 + 0.9 * -10 = -10$$

$$Q(S, I) = 1 + 0.9 * 10 = 10$$

$$Q(L, B) = 1 + 0.9 * 10 = 10$$

$$Q(S, B) = -1 + 0.9 * -10 = -10$$

Conclusion : For state S action I is best and for state L both actions B and P are best.

Policy Improvement II

$\pi_2(L)$ = P and then I

$\pi_2(S)$ = I

$$V_{\pi_2}(L) = 1 + 0.9 * 10 = 10$$

$$V_{\pi_2}(S) = 1 + 0.9 * 10 = 10$$

Q-Update :

$$Q(L, N) = -1 + 0.9 * 10 = 8$$

$$Q(S, N) = -1 + 0.9 * 10 = 8$$

$$Q(L, P) = 1 + 0.9 * 10 = 10$$

$$Q(S, P) = -1 + 0.9 * 10 = 8$$

$$Q(L, I) = -1 + 0.9 * 10 = 8$$

$$Q(S, I) = 1 + 0.9 * 10 = 10$$

$$Q(L, B) = 1 + 0.9 * 10 = 10$$

$$Q(S, B) = -1 + 0.9 * 10 = 8$$

Conclusion : Though the values of expected reward changed the best policy is the same as above i.e. $\pi_3(L)$ = P and then I

$\pi_3(S)$ = I

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

Solution: Given that initially, while he was writing this question, house is in Laughter state so we should play organ and when it stops we should burn incense. We are following the policy we obtained in above problem.

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: Given that action is applied at t + τ so the reward and next state is obtained by next time step so we can denote that reward as $R_{t+\tau+1}$ here 1 is the time step, this reward is result of action, control agent taken at t. Return -

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

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

Solution:

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