CS6700 : Reinforcement Learning
Written Assignment #1

Topics: Intro, Bandits, MDP, Q-learning, SARSA **Deadline**: 11 March 2022, 11:55 pm
Name: Rongali Rohith **Roll number:** EE19B114

- 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. (5 marks) [MDP as Bandit] Consider the following grid world task. The environment is a 10 × 10 grid. The aim is to learn a policy to go from the start state to the goal state in the fewest possible steps. The 4 deterministic actions available are to move one step up, down, left or right. Standard grid world dynamics apply. The agent receives a reward of 0 at each time step and 1 when it reaches the goal. There is a discount factor $0 < \gamma < 1$. Formulate this problem as a family of bandit tasks. These tasks are obviously related to one another. Describe the structure of the set up and the rewards associated with each action for each of the tasks, to make it perform similarly to a Q-learning agent.

Solution: Assume that reward for each state is -1 and goal state is +1.

We would consider a total of 100 bandits, mapped to each state in the grid world. Each of the bandit should have 4 arms corresponding to actions in the grid world except the ones at the edges(3 arms) and corners(2 arms). In a grid world we move from one state to another based on the actions given by the policy(which can be ϵ-greedily chosen). In this bandit-formulation too we move from a bandit to its neighbouring bandits based on the arm chosen(i.e. the action it indicates).

The reward corresponding to bandit-arms is assumed to be equivalent to the return that we define in case of a grid world.Now our problem is to figure out the arm/arms for each of the 99 bandits(excluding end-state) that maximises this return.

To perform like a Q-learning agent we consider the action value($Q_{ij}(a)$) for each bandit to be same as state-action value of the corresponding state. And we perform update(similar to Q-learning) as follows:

$$Q_{ij}(a) \leftarrow Q_{ij}(a) + \alpha[r + \gamma max_{a'} Q_{i'j'}(a') - Q_{ij}(a)]$$

As in Q-learning we would estimate these bandit action values over various trajectories. From these action value estimates we would be able to figure out optimal arms for each of the bandits.

2. (4 marks) [Delayed reward] 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?

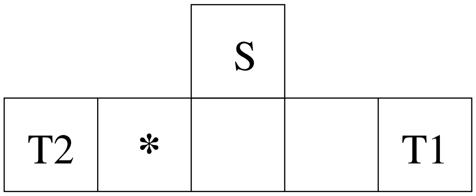
   Solution:
   Return: $G_t = R_{t+\tau+1} + \gamma R_{t+\tau+2} + \gamma^2 R_{t+\tau+3} + ...$

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

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

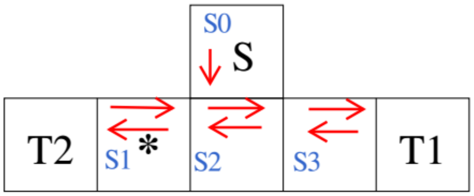
3. (5 marks) [Blackwell Optimality] For some policy π in an MDP, if there exists a constant k, such that for all $\gamma \in [k, 1)$, π is optimal for the discounted reward formulation, then π is said to be *Blackwell optimal.* Consider the gridworld problem shown below, where S denotes a starting state and T1 and T2 are terminal states. The reward for terminating in T1 is +5 and for terminating in T2 is +10. Any transition into the state marked $*$ has a reward of $a \in (-\infty, 0)$. All other transitions have a reward of 0.

   For this problem, give a characterization of Blackwell optimal policies, in particular the value k, parameterized by a. In other words, for different ranges of a, give the Blackwell optimal policy, along with the value of k.



Solution: The above figure indicates the different states and the possible actions in them. Lets consider all the deterministic policies, there are 2^3 different policies obtained by considering combinations of left or right actions in the states S1,S2 and S3.

Denoting action left by 0 and action right by 1, each policy is determined by a 3-digit



binary number. For example $\pi = 100$ denotes the policy in which $\pi(S1) = right$, $\pi(S2) = left$ and $\pi(S3) = left$. Now we compute the value functions corresponding to all the 8 policies:

$\pi_1 = 000$:
$V(S0) = a\gamma + 10\gamma^2$, $V(S1) = 10$, $V(S2) = A + 10\gamma$, $V(S3) = a\gamma + 10\gamma^2$

$\pi_2 = 001$:
$V(S0) = a\gamma + 10\gamma^2$, $V(S1) = 10$, $V(S2) = A + 10\gamma$, $V(S3) = 5$

$\pi_3 = 010$:
$V(S0) = 0$, $V(S1) = 10$, $V(S2) = 0$, $V(S3) = 0$

$\pi_4 = 011$:
$V(S0) = 5\gamma^2$, $V(S1) = 10$, $V(S2) = 5\gamma$, $V(S3) = 5$

$\pi_5 = 100$:
$V(S0) = \frac{a\gamma}{1-\gamma^2}$, $V(S1) = \frac{a\gamma}{1-\gamma^2}$, $V(S2) = \frac{a}{1-\gamma^2}$, $V(S3) = \frac{a\gamma}{1-\gamma^2}$

$\pi_6 = 101$:
$V(S0) = \frac{a\gamma}{1-\gamma^2}$, $V(S1) = \frac{a\gamma}{1-\gamma^2}$, $V(S2) = \frac{a}{1-\gamma^2}$, $V(S3) = 5$

$\pi_7 = 110$:
$V(S0) = 0$, $V(S1) = 0$, $V(S2) = 0$, $V(S3) = 0$

$\pi_8 = 111$:
$V(S0) = 5\gamma^2$, $V(S1) = 5\gamma^2$, $V(S2) = 5\gamma$, $V(S3) = 5$

For $a \in (-5, 0)$:
π_1 is Blackwell optimal for $k = -\frac{a}{5}$

For $a \in (-\infty, -5]$:
π_4 is Blackwell optimal for k=0

4. (10 marks) [Jack's Car Rental] Jack manages two locations for a nationwide car rental company. Each day, some number of customers arrive at each location to rent cars. If Jack has a car available, he rents it out and is credited \$ 10 by the national company. If he is out of cars at that location, then the business is lost. Cars become available for renting the day after they are returned. To help ensure that cars are available where they are needed, Jack can move them between the two locations overnight, at a cost of \$ 2 per car moved. We assume that the number of cars requested and returned at each location are Poisson random variables, meaning that the probability that the number n is $\frac{\lambda^n}{n!}e^{-\lambda}$, where λ is the expected number. Suppose λ is 3 and 4 for rental requests at the first and second locations and 3 and 2 for returns. To simplify the problem slightly, we assume that there can be no more than 20 cars at each location (any additional cars are returned to the nationwide company, and thus disappear from the problem) and a maximum of five cars can be moved from one location to the other in one night.

   (a) (4 marks) Formulate this as an MDP. What are the state and action sets? What is the reward function? Describe the transition probabilities (you can use a formula rather than a tabulation of them, but be as explicit as you can about the probabilities.) Give a definition of return and describe why it makes sense.

Solution:
State set: (n_1, n_2) where n_1 and n_2 are the cars in locations 1 and 2.

Action is defined by a = net number of cars moved from location-1 to location-2. Range of a=[-5,5]

In the next day, let $rn1, rn2$ denote the number of cars rented out in location 1 and 2 while $rt1, rt2$ denote the number of cars returned in location 1 and 2.
Reward $= -2|a| + 10(rn1 + rn2)$

Now at the end of the day we move into new state (n_1', n_2')
Computing the transition probability $(n_1, n_2) \to (n_1', n_2')$:
Note that the value of 'a' should be such that $n_1 - a > 0$ and $n_2 + a > 0$, only then will it lead to valid states. Also there is a cap of 20 cars so $min(n_1 - a, 20)$ and $min(n_2 + a, 20)$ is to be considered instead.
$rt1 - rn1 = n_1' - min(n_1 - a, 20) = x1$
$rt2 - rn2 = n_2' - min(n_2 + a, 20) = x2$

$p((n'_1, n'_2)/(n_1, a, n_2)) = p(n'_1/(n_1, a, n_2))p(n'_2/(n_1, a, n_2))$
$p(n'_1/(n_1, a, n_2)) = \sum_{rn1=0}^{min(n_1-a,20)} \frac{3^{rn1}e^{-3}}{(rn1)!} \frac{3^{x_1+rn1}e^{-3}}{(x_1+rn1)!}$
$p(n'_2/(n_1, a, n_2)) = \sum_{rn2=0}^{min(n_2+a,20)} \frac{4^{rn2}e^{-4}}{(rn2)!} \frac{2^{x_2+rn2}e^{-2}}{(x_2+rn2)!}$
The above formula works only if these conditions hold: $0 < n'_1 < 20$, $0 < n'_2 < 20$.

If any one of n'_1 or n'_2 is equal to 20 implies that there is a chance that at the end of the day, there could have been more than 20 cars but the excess ones(how many we don't know) have been returned to company. There may not be a bound for the number of returns on that day. For now let me consider the case of $n'_1 = 20$, other ones can be written similarly.
Let's characterise the excess returns in terms of ϵ:
$p(n'_1 = 20/(n_1, a, n_2)) = \sum_{\epsilon=0}^{\infty} \sum_{rn1=0}^{min(n_1-a,20)} \frac{3^{rn1}e^{-3}}{(rn1)!} \frac{3^{x_1+rn1+\epsilon}e^{-3}}{(x_1+rn1+\epsilon)!}$

We could characterise the return as
$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ...$

We don't want any location to run out cars, this could be modelled by giving a high negative reward for such states. Or even without this high negative reward, the return equation that we have will be maximised when we can keep the company in business for long time. Discounted return may be helpful in preventing such 'out of business' state by inducing a sense of future reward to the agent i.e. we don't want the business to terminate very early(leads to low returns).

(b) (3 marks) One of Jack's employees at the first location rides a bus home each night and lives near the second location. She is happy to shuttle one car to the second location for free. Each additional car still costs \$ 2, as do all cars moved in the other direction. In addition, Jack has limited parking space at each location. If more than 10 cars are kept overnight at a location (after any moving of cars), then an additional cost of \$ 4 must be incurred to use a second parking lot (independent of how many cars are kept there). These sorts of nonlinearities and arbitrary dynamics often occur in real problems and cannot easily be handled by optimization methods other than dynamic programming. Can you think of a way to incrementally change your MDP formulation above to account for these changes?

Solution:
Incorporating the employee's help in shuttling one car from location 1 to 2 we get reward function as:

$R = -2a' + 10(rn1 + rn2)$
where $a' = min(|a|, |a-1|)$
The extra cost for second parking lot can be viewed in terms of reduction in the reward as follows. Note that $|a|$ cars are being shuttled and hence subtracted from x_1 or x_2 depending on the sign of a:
if $(x_1 - max(a, 0)) > 10$: R ← R-4
if $(x_2 + min(a, 0)) > 10$: R ← R-4

(c) (3 marks) Describe how the task of Jack's Car Rental could be reformulated in terms of *afterstates*. Why, in terms of this specific task, would such a reformulation be likely to speed convergence? *(Hint:- Refer page 136-137 in RL book 2nd edition. You can also refer to the video at https://www.youtube.com/watch?v=w3wGvwi336I)*

Solution: In this case the after-states can be considered to be the number of cars in location-1$(n_1 - a)$ and location-2$(n_2 + a)$ after the shuttling is done. Also different pairs of (n_1, a) and (n_2, a) leading to the same after-state(of the number cars after shuttling) will now be given the same value. For this specific task it is likely to speed up convergence because here we need to estimate a simpler mapping i.e. we need not to worry about the car transfer number 'a'.

5. (7 marks) [Organ Playing] 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) (4 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 set = (Laughter, Quiet)
Denote Burn incense as 'BI' and organ playing as 'O':
Action set = (BI, O, BI and O, neither)
The following table gives the state transitions and rewards:

Present state	action	next state	Reward
Laughter	BI	Laughter	-1
Laughter	O	Quiet	+1
Laughter	BI and O	Quiet	+1
Laughter	neither	Laughter	-1
Quiet	BI	Quiet	+1
Quiet	O	Laughter	-1
Quiet	BI and O	Laughter	-1
Quiet	neither	Laughter	-1

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

Solution: Denote Laughter state by 'L' and Quiet state by 'Q' Given initial policy π_i such that $\pi_i(L) = BI$ $\pi_i(Q) = BI$.
V(L) = V(Q) = 0
Policy evaluation
Assume $\theta = 0.6$

$V(L) = -1 + 0.90 = -1$, $V(Q) = +1 + 0.90 = +1$, $\Delta V = 1$

$V(L) = -1 + 0.9(-1) = -1.9$, $V(Q) = +1 + 0.9(1) = +1.9$, $\Delta V = 0.9$

$V(L) = -1 + 0.9(-1.9) = -2.71$, $V(Q) = +1 + 0.9(1.9) = +2.71$, $\Delta V = 0.81$

$V(L) = -1 + 0.9(-2.71) = -3.439$, $V(Q) = +1 + 0.9(2.71) = +3.439$, $\Delta V = 0.729$

$V(L) = -1 + 0.9(-3.439) = -4.0951$, $V(Q) = +1 + 0.9(3.439) = +4.0951$, $\Delta V = 0.6561$

$V(L) = -1 + 0.9(-4.0951) = -4.68559$, $V(Q) = +1 + 0.9(4.0951) = +4.68559$, $\Delta V = 0.59049$

Policy improvement:
$\pi(s) \leftarrow Argmax_a \Sigma_{s'} p(s'/s, a)[E[r/s, a, s'] + \gamma V(s')]$

$\pi(L) = $ O or (BI and O)
$\pi(Q) = $ BI

STEP-2:
Policy evaluation:
$V(L) = +1 + 0.9(4.68559) = +5.217031$, $V(Q) = +1 + 0.9(4.68559) = +5.217031$, $\Delta V = 0.53$

Policy improvement:
$\pi(L) = $ O or (BI and O)
$\pi(Q) = $ BI
Policy is stable.

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

Solution: When there is laughter play organ post which it becomes silent, then continue burning incense(and don't play organ) for rest of the time to keep the room quiet.

6. (4 marks) [Stochastic Gridworld] An ϵ-greedy version of a policy means that with probability 1-ϵ we follow the policy action and for the rest we uniformly pick an action. Design a stochastic gridworld where a deterministic policy will produce the same trajectories as a ϵ-greedy 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 the stochastic grid world, the agent moves in the direction of chosen action with probability of $1 - \frac{3\epsilon}{4}$ and moves in either west,east or south of the chosen direction with equal probability of $\frac{\epsilon}{4}$.

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

Solution: Yes, it would converge to the same policy in the two worlds.
Note that the underlying MDP model(and hence the bellman equation) should be same for both the worlds(i.e. in terms of transition probabilities, expected

reward). Therefore, Sarsa on both the worlds would estimate the same action-value functions and thus they would converge to the same policy.

7. (5 marks) [Contextual Bandits] Consider the standard multi class classification task (Here, the goal is to construct a function which, given a new data point, will correctly predict the class to which the new point belongs). Can we formulate this as contextual bandit problem (Multi armed Bandits with side information) instead of standard supervised learning setting? What are the pros/cons over the supervised learning method. Justify your answer. Also describe the complete Contextual Bandit formulation.

Solution: Yes, the multi-class classification problem can be formulated as a contextual bandit problem. As an example consider the image-classification problem where we want to classify the given image among 'n' classes. The image(in terms of its features) would serve as a context to the bandit while there would be n-arms corresponding to each class, we have to figure out which arm shall be pulled in order to maximise reward(in this case a positive reward if it identifies the correct class and reward of zero if it identifies the wrong class). After sufficient iterations, the model will be able to figure out the right arm for each context.

pros : In standard supervised learning, we have training data on which our model is trained and then tested and used to predict and not allowed to change later. Whereas in the case of contextual bandits the learning happens in an online fashion improving itself with each iteration.

cons:Unlike supervised learning we may not have a very good classification model in the very beginning. The model needs to observe many context vectors before it can achieve comparable accuracy.