CS6700 : Reinforcement Learning
Written Assignment #1

Topics: Intro, Bandits, MDP, Q-learning, SARSA **Deadline**: 11 March 2022, 11:55 pm
Name: Vishal Rishi MK **Roll number:** CH18B013

- 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: We assume a standard deterministic grid world. We can view this as a multi-arm bandit formulation. At each state s, let $|A|$ denote the cardinality of the action set which is also equal to the number of arms. This gives us 100 (10x10) bandits in total (one for each state). Let $Q_k(s,a)$ denote the action value of the bandit in state s for picking an arm a at the kth time step. When the bandit at state s picks an arm a, it receives a reward of γ^{n-1}, where n denotes the number of time steps to reach the goal state after picking arm a. The action value is updated as,

$$Q_{k+1}(s,a) \;=\; Q_k(s,a) \;+\; \tfrac{1}{k}[\gamma^{n-1} \;-\; Q_k(s,a)]$$

The behavioural policy is ϵ-greedy with respect to the action values at each state. Formally,

$$\pi(a|s) \;=\; \begin{cases} 1-\epsilon+\frac{\epsilon}{|A|}, & a = argmax\, Q(s,arm) \\ \frac{\epsilon}{|A|}, & a \neq argmax\, Q(s,arm) \end{cases}$$

With an initialisation of the action values for all the bandits, we pick actions according to the behavioural policy. Once the goal state is reached, we update the action values for the bandits. This is repeated till convergence. We point out that this bandit formulation is similar in principle to Monte-Carlo control.

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: The control agent takes an action on observing the state at time t. Since the actions are applied to the system at time $t + \tau$, we receive rewards due to the action only at time $t + \tau$. Hence, an appropriate notion of return for this task would be,

   $G_t = R_{t+\tau+1} + \gamma R_{t+\tau+2} + \gamma^2 R_{t+\tau+3} + ... = R_{t+\tau+1} + \gamma G_{t+1}$
   where γ is the discount factor.

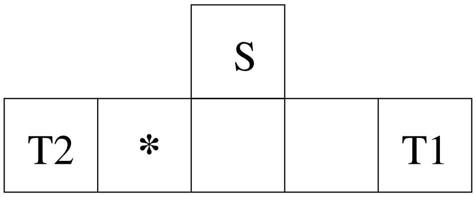
   (b) (2 marks) Give the TD(0) backup equation for estimating the value function of a given policy.

   Solution: We define $\nu_\pi(s) = E_\pi [G_t | S_t = s]$. The TD(0) backup equation for estimating the value function becomes,

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

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: For this grid, we have two policies π_1 and π_2. The policy π_1 always leads to the terminal state T2 and the policy π_2 always leads to the terminal state T1. Using the Bellman equations, we compute the value function at the starting state S under both the policies.

$\nu_{\pi_1}(S) \;=\; a\gamma \;+\; 10\gamma^2$
$\nu_{\pi_2}(S) \;=\; 5\gamma^2$
Since S is the starting state, an optimal policy would have a greater value function at S. We can see that when $a \leq -5$, $\nu_{\pi_2}(S) \;\geq\; \nu_{\pi_1}(S)$ for all values of $\gamma \in [0, 1]$. But when $a > -5$, $\nu_{\pi_1}(S) \;\in\; \nu_{\pi_2}(S)$ when $\gamma \in [-a/5, 1)$. In other words, the Blackwell optimal policy, $\pi_{blackwell}$ is given by,

$$\pi_{blackwell} \;=\; \begin{cases} \pi_2, & a \leq -5,\ k = 0 \\ \pi_1, & a > -5,\ k = -a/5 \end{cases}$$

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: At the start of a particular day, let x_i denote the number of cars that can be rented at location i. Let $rent_i$ denote the number of cars that were rented on a particular day at location i. Let acc_i denote the number of cars that were accommodated on a particular day at location i. Let c be the number of cars that were transported from location 2 to location 1. We define state s as follows:
$s \; \equiv \; (x_1, rent_1, acc_1, x_2, rent_2, acc_2)$

where $0 \leq x \leq 20, \; 0 \leq rent \leq x, \; 0 \leq acc \leq (20 - x + rent)$
The action set is,
$max(-5, -x_1 - \Delta_1, x_2 + \Delta_2 - 20) \leq c \leq min(5, 20 - x_1 - \Delta_1, x_2 + \Delta_2)$
with $\Delta \; = \; acc - rent$
Let λ_{r_i} denote average number of cars requested at location i. Let λ_{a_i} denote average number of cars returned at location i. We define

$$p_i(rent) \; = \; \begin{cases} \lambda_{r_i}^{rent} e^{-\lambda_{r_i}}/rent!, & rent < x \\ \sum_{k=x}^{\infty} \lambda_{r_i}^{k} e^{-\lambda_{r_i}}/k!, & rent = x \end{cases}$$

$$q_i(acc) \; = \; \begin{cases} \lambda_{a_i}^{acc} e^{-\lambda_{a_i}}/acc!, & acc < 20 - x + rent \\ \sum_{k=20-x+rent}^{\infty} \lambda_{a_i}^{k} e^{-\lambda_{a_i}}/k!, & acc = 20 - x + rent \end{cases}$$

at location i.
We define the state transition probabilities as follows:

$$p(s'|s, c) = \begin{cases} p_1(rent'_1)p_2(rent'_2)q_1(acc'_1)q_2(acc'_2), & x'_1 = x_1 + \Delta_1 + c, x'_2 = x_2 + \Delta_2 - c \\ 0, & otherwise \end{cases}$$

where $s' \; \equiv \; (x'_1, rent'_1, acc'_1, x'_2, rent'_2, acc'_2)$
The reward function is as follows:

$$E(r|s, c, s') \; = \; 10(rent'_1 + rent'_2) \; - \; 2|c|$$

The definition of return would be

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

This would be an infinite sum because of the non-episodic nature of the MDP.

(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:

(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: We can focus the after-states about parking numbers at each day's morning. It will be the result of yesterday's actions. It will decrease the computation because we have many different ways of arranging the cars but will end up next morning the same number of cars at each location. After-states are useful when we have knowledge of an initial part of the environment's dynamics but not necessarily of the full dynamics. For example, in games we typically know the immediate effects of our moves. We know for each possible chess move what the resulting position will be, but not how our opponent will reply. After-state value functions are a natural way to take advantage of this kind of knowledge and thereby produce a more efficient learning method.

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: l and nl (laughing and non-laughing states, respectively)
Action set: o and i (playing the organ and burning incense, respectively)
The following are the state transition probabilities:

$P(nl|l,o) \;=\; 1$ and $P(l|l,i) \;=\; 1$
$P(l|nl,o) \;=\; 1$ and $P(nl|nl,i) \;=\; 1$

Reward function:

$$R(s,a,s') \;=\; \begin{cases} +1, & s' = nl \\ -1, & s' = l \end{cases}$$

$R(s,a,s')$ is the expected reward from the transition (s,a,s').

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

Solution: Let π_0 be the initial policy. $\pi_0(l) = i$ and $\pi_0(nl) = i$.
We evaluate π_0 using the Bellman equations. This gives,
$\nu_{\pi_0}(l) \;=\; -10$ and $\nu_{\pi_0}(nl) \;=\; 10$
To improve π_0, we compute the state-action values.
$q_{\pi_0}(l,i) \;=\; -10$ and $q_{\pi_0}(l,o) \;=\; 10$
$q_{\pi_0}(nl,o) \;=\; -10$ and $q_{\pi_0}(nl,i) \;=\; 10$
From the state-action values, we compute π_1 such that
$\pi_1(s) \;=\; argmax\, q_{\pi_0}(s,a)$.
Thus $\pi_1(l) = o$ and $\pi_1(nl) = i$. We evaluate π_1 using the Bellman equations. This gives,
$\nu_{\pi_1}(l) \;=\; 10$ and $\nu_{\pi_1}(nl) \;=\; 10$
To improve π_1, we compute the state-action values.
$q_{\pi_1}(l,i) \;=\; 8$ and $q_{\pi_1}(l,o) \;=\; 10$
$q_{\pi_1}(nl,o) \;=\; 8$ and $q_{\pi_1}(nl,i) \;=\; 10$
Improving upon π_1 gives π_2 such that $\pi_2(l) = o$ and $\pi_2(nl) = i$. We can see that $\pi_1 = \pi_2$. Hence, $\nu_{\pi_1} \;=\; \nu_{\pi_2}$. Thus $\pi_1 = \pi_2$ is the optimal policy, according to policy improvement theorem.

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

Solution: From the optimal policy, one should play the organ when we hear laughter. Once the laughter stops, we should burn incense.

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: Let $O(s, s')$ be the action a that makes a transition from state s to state s' in a completely deterministic standard grid world. If $O(s, s')$ is NULL, we say that the state s' is not reachable from state s in a single transition. Let $\pi(s)$ be a deterministic policy. The ϵ-greedy policy built from $\pi(s)$ is given by,

$$\pi'(a|s) \;=\; \begin{cases} 1-\epsilon+\frac{\epsilon}{|A|}, & a=\pi(s) \\ \frac{\epsilon}{|A|}, & a\neq\pi(s) \end{cases}$$

   where $|A|$ is the cardinality of the action set in state s. We can define a stochastic grid world, where the transition probabilities are given by,

$$p(s'|s,a) \;=\; \begin{cases} 1-\epsilon+\frac{\epsilon}{|A|}, & a=O(s,s') \\ \frac{\epsilon}{|A|}, & a\neq O(s,s') \\ 0, & O(s,s')\ is\ null \end{cases}$$

   such that for a given trajectory, its probability in a standard deterministic world under an ϵ-greedy policy is the same as its probability in the stochastic grid world under a deterministic policy. The reward function, state and action sets of the stochastic world are similar to that of the standard deterministic grid world.

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

   Solution: We assume that the expected rewards for all the transitions in both the grid worlds are the same. But the transition probabilities differ for both the worlds. Hence, the optimal value function might be different for both the worlds (according to the Bellman optimality equation). Hence SARSA might converge to different value functions when trained on the two grid worlds. This might give rise to completely different optimal policies.

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, we can formulate the supervised multi-class classification task as a contextual multi-arm bandit problem. The context in the bandit formulation, s, would be the feature vector $\mathbf{x}$ in the classification problem. The number of arms in the bandit formulation would be the number of classes. Given the context s, the agent picks an arm (class). If the arm corresponds to the target class of s, we provide a positive reward. Else, the agent receives a negative reward. The agent might struggle to distinguish between classes when there is class imbalance. Particularly, when we have very few data points for a class, the agent would not have learned the optimal policy.