Reinforcement Learning Mid-term Exam

Note 1: All the answers should be typed in MS-word or Latex and the pdf file to be submitted.

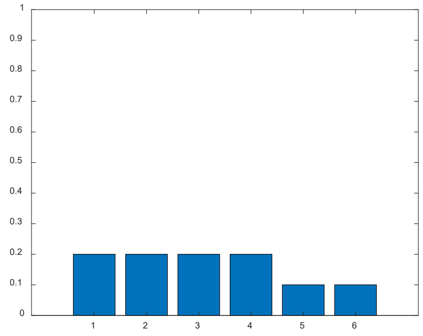
(No handwritten answers are accepted)

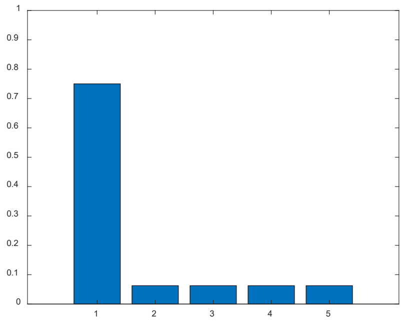
Note 2: correct numerical answer worth 50 percent of the point for each question. This means that if you get the final answer wrong but your approach is correct, you will loose 50 percent of the point. So double check your final answer!

Note 3: Total points for this exam is 100.

1- Consider two random variables with distributions below:

$$p = \{0.2, 0.2, 0.2, 0.2, 0.1, 0.1\}$$
$$q = \{0.75, 0.0625, 0.0625, 0.0625, 0.0625\}$$





- [4 points] Calculate the entropy for each variable.
- [4 points] Intuitively how can you tell which variable has a higher entropy without calculating the entropy numerically? What does higher entropy mean?

2- [5 points] Which equation correctly relates v_* to q_* ? draw the corresponding backup-diagram and explain.

A. $v_*(s) = \sum_{r,s} \pi(a|s)p(s',r|s,a)[r + \gamma q_*(s')]$

B. $v_*(s) = \max_a q_*(s,a)$

C. $v_*(s) = \max_a \sum_{r,s} p(s',r|s,a)[r + \gamma q_*(s',a)]$

D. $v_*(s) = \max_a \sum_{r,s'} p(s',r|s,a)[r + \gamma v_*(s')]$

E. None of the above

F. B and D

3- Consider a vehicle with 4 actions (left,right,up, down). There’s no uncertainty in the outcome of the action (i.e. when left is commanded, the left state is achived). The actions that cause the vehicle outside the grid, leave the state unchanged.

The reward for all transition is -1 except when the goal is reached where the reward is zero. Discount factor $\gamma = 1$.

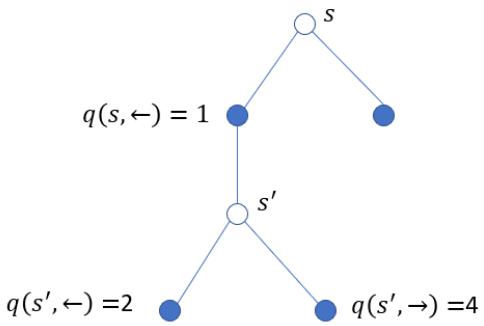
The figure on left shows the name of the states and figure on the right shows the state-value $V(s)$, for each state under a uniform random policy.

Termination	a	b	c
d	e	f	g
h	i	j	k
l	m	n	Termination

Termination	-14	-20	-22
-14	-18	?	-20
-20	-20	-18	-14
-22	-20	-14	Termination

- [4 points] What is $q(k, \downarrow)$?
- [4 points] What is $q(g, \downarrow)$?
- [4 points] What is V(f) ?

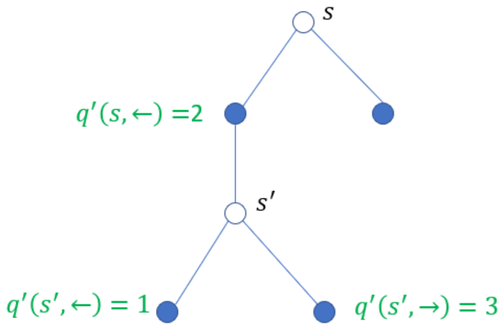
4- Consider the state-action interaction below. The $q(s,a)$ written next to each action (left and right) is the initial estimate.



consider discount factor $\gamma = 0.5$ and learning rate $\alpha = 0.1$:
What is the target value as well as updated value of $q(s,\leftarrow)$ using SARSA algorithm if the action at s' was the left action.

- [4 points] What is the target value as well as updated value of $q(s,\leftarrow)$ using SARSA algorithm if the action at s' was the right action.
- [4 points] Assume that the action at s' has a distribution in such a way that it is 30 percent left action and 70 percent right action. What is the expected SARSA target value and as well expected value of $q(s,\leftarrow)$ under SARSA algorithm?
- [4 points] What is the target value as well as updated value of $q(s,\leftarrow)$ using Q-learning algorithm.
- [4 points] Does the distribution of the action at s' have an effect on the Q-learning target value?

Consider now we have another batch of initial estimates for the action-values. We call then $q'(s,a)$. They are shown in the diagram below in green color:



- [4 points] What is the updated value of $q(s,\leftarrow)$ using double-Q learning algorithm? (You need to use both green and black initial estimates)

- [4 points] What is the updated value of $q'(s, \leftarrow)$ using double-Q learning algorithm? (You need to use both green and black initial estimates)

5- As we discussed, n-step off-policy return via bootstrapping can be written as:

$$G_{t:h} = \rho_t (R_{t+1} + \gamma G_{t+1:h}) \quad (1)$$

where $\rho_t = \frac{\pi(A_t|S_t)}{b(A_t|S_t)}$ is the importance sampling ratio between target and behavior policy.

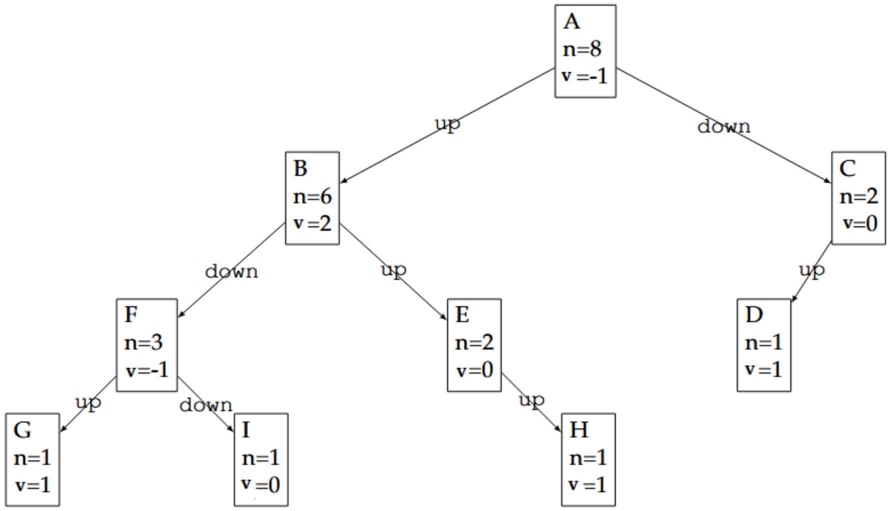
Using control variates, the return can be written as:

$$G_{t:h} = \rho_t (R_{t+1} + \gamma G_{t+1:h}) + (1 - \rho_t) V_{h-1}(S_t) \qquad (2)$$

- [8 points] Prove that introducing the control variates in equation (2) does not add any bias to the original return (i.e. equation (1)) in expectation.

6- [8 points] If you were to design an algorithm called TD-search by modifying the Monte-Carlo Tree Search, how do you go about it. In particular, which part of the MCTS you would modify and how?

7- Assume we are using Monte Carlo Tree Search (MCTS) to decide on the next action for a game with two actions at each state (up and down). The state of the tree at time t is as follows:



Each state has a name (A, B, ..), a return value v and n value. Assume that the constant *c* in UCB1 is 0.5.

- [5 points] What is the node that is selected next? Show your work.
- [2 points] What action is next for the selected node? This action will make a new state call it "J".
- [2 points] What is the "n" and "v" value for the new state "J"?
- [5 points] if the simulation rollout from the expanded node gives a value of 1, backup that value to all of the affected nodes.
- [4 points] assume that after this final rollout, we've run out of time to run the MCTS simulation and must now choose an action from state A. What action will be chosen and why if we use the greedy tree policy.

8- In this question we are comparing mean and variance of the estimate reward for on-policy and off-policy algorithms. Consider the on-policy scenario below where actions are taken under target policy π:

Action	Reward	Probability of action under policy π
→	+10	0.9
←	+20	0.1

- [2 points] What is the expected value (mean) of estimated reward?
- [4 points] What is the variance of the estimate reward?
  Hint: use the equation $Var[X] = \mathbb{E}[X^2] - \mathbb{E}[X]^2$

Now assume that there is a behavior policy that is taking actions with following distribution:

Action	Reward	Probability of action under target policy π	Probability of action under behavior policy b
→	+10	0.9	0.5
←	+20	0.1	0.5

- [3 points] Assume that we don't know the distribution of the target policy and we only know the ratio $\frac{\pi}{b}$ (importance sampling ratio).
  Numerically calculate the expected reward assuming that the action are taken by the behavior policy.
- [5 points] Calculate the variance of the expected reward?
- [3 points] What is the intuition behind higher variance in off-policy in this example? Can you explain how the importance sampling could increase the variance in off-policy learning?