CCBR 2020 : Introduction to Reinforcement Learning

Assignment #1 **Max marks: 50**

Deadline: 7th February 2020 23:55 hrs

Instructions:

1. This is an individual assignment. Collaborations and discussions are strictly prohibited.
2. **Any sort of plagiarism/cheating will be dealt very strictly. Acknowledge any source used.**
3. You have to turn in the well-documented code along with a detailed report of the results of the experiments.
4. Be precise with your explanations. Avoid verbosity. Place relevant results that bolster your conclusions. Report should be **within 10 pages** in a single column format, and with 11pt font. Results/conclusions beyond the page limit will be ignored during evaluation.
5. You can use any programming language for this assignment. However, we recommend Python or MATLAB.
6. Please email your report and code to TAs (Nirav Bhavsar, Ajay Pandey).

1 The Quiz Problem

Given a list of N questions. If question i is answered correctly (given probability p_i), we receive reward R_i; if not the quiz terminates. Find the optimal order of questions to maximize expected reward. (*Hint: Optimal policy has an "index form".*) (5 marks)

2 Finite Horizon MDP

For the finite horizon MDP, consider the following alternative cost objective for any policy π and initial state x_0 :

$$J_\pi(x_0) = \mathbb{E}\left[\exp\left(g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, \mu_k(x_k), x_{k+1})\right)\right]$$

Answer the following: (3 + 2 marks)

(a) Show that an optimal cost and an optimal policy can be obtained by the following DP-algorithm variant:

$$J_N(x_N) = \exp(g_N(x_N))$$
$$J_k(x_k) = \min_{a_k \in A(x_k)} \mathbb{E}_{x_{k+1}}\left(\exp(g_k(x_k, a_k, x_{k+1}))\, J_{k+1}(x_{k+1})\right)$$

(b) Let $V_k(x_k) = \log J_k(x_k)$. Assume that the single stage cost g_k is a function of x_k and a_k only (and does not depend on x_{k+1}. Then, show that the DP algorithm, which is specified above, can be re-written as

$$V_N(x_N) = g_N(x_N)$$
$$v_k(x_k) = \min_{a_k \in A(x_k)} \left(g_k(x_k, a_k) + \log \mathbb{E}_{x_{k+1}}\left(\exp(V_{k+1}(x_{k+1}))\right)\right.$$

3 The Food Buying Problem

You are walking along a line of N stores in ashopping complex, looking to buy food before entering a movie hall at the end of the store line. Each store along the line has a probability p of providing the food you like. You cannot see what the next store (say $k+1$) offers, while you are at the k th store and once you pass store k, you cannot return to it. You can choose to buy at store k, if it has the food item you like and pay an amount $N-k$ (since you have to carry this item for a distance proportional to $N-k$). If you pass through all the stores without buying, then you have to pay $\frac{1}{1-p}$ at the entrance to the movie hall to get some food.

Answer the following: (2+2+4 marks)

(a) Formulate this problem as a finite horizon MDP.

(b) Write a DP algorithm for solving the problem.

(c) Characterize the optimal policy as best as you can. This may be done with or without the DP algorithm.

Hint:

- *Follow the technique from the asset selling example (see Section 4.4 in D.P.Bertsekas, Dynamic Programming and Optimal Control, Vol. I, 3^{rd} edition, Athena Scientific, 2005), since the question is about when to stop.*
- *Argue that if it is optimal to buy at store k when the item you like is available, then it is optimal to buy at store $k+1$ when the likeable food is available there.*
- *Show that $\mathbb{E}\left(J_{k+1}(x)\right)$ is a constant that depends on $n-k$ leading to an optimal threshold-based policy.*

4 Discounted Cost MDP

Consider a discounted cost MDP with two states, denoted A and B, specified through the transition diagram below, where the edge labels are in the following format: (action (a), single-stage cost g(i, a, j), probability $p_{ij}(a)$).

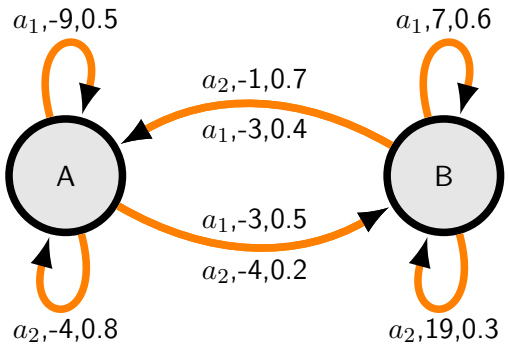

Figure 1: MDP

Assume that the discount factor α is set to 0.9 and answer the following: (3 + 4 + 4 + 2 marks)

(a) Find the optimal policy for this problem.

(b) Start with a policy that chooses action a_1 in each state, and perform policy iteration.

(c) Start with the zero vector, and perform value iteration for four steps. Show the cost vector and the corresponding policy in each step.

(d) Does the optimal policy change, when the discount factor is 0.1? Justify your answer.

5 Value and Policy Iteration (Programming)

The objective of this question is to compare value iteration and policy iteration on a 10×10 gridworld based on the actions, rewards and the state space given below.

- **State space**: Gridworld has 100 distinct states. There are two variants of this gridworld, one with a terminal state as Goal 1 and other with Goal 2. For the variant with Goal 1 as a terminal state, Goal 2 is treated as a normal state and vice-versa. There are two wormholes labeled as IN in Grey and Brown, any action taken in those states will teleport you to state labeled OUT in Grey and Brown respectively. In case of Grey wormhole you can teleport to any one of the states labelled OUT with equal probability (i.e. 1/4). States labeled OUT is just a normal state. An instance of this gridworld is shown in the figure below.

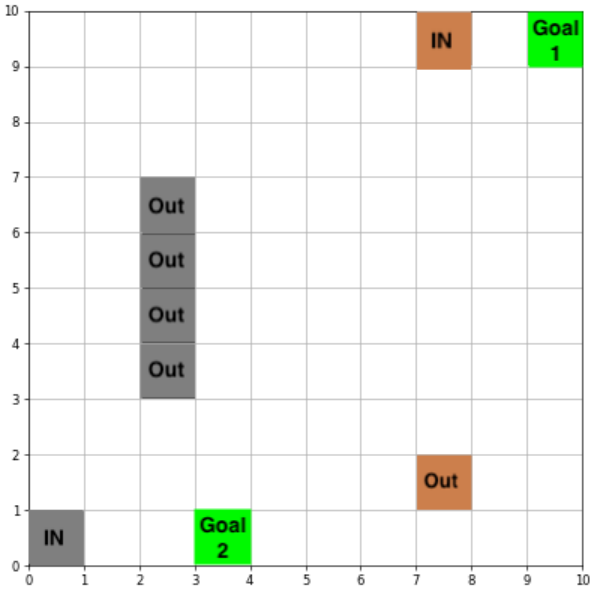

Figure 2: 10×10 gridworld

- **Actions:** In each non-terminal state, you can take 4 actions $\mathcal{A} = \{\text{Up, Down, Left, Right}\}$, which moves you one cell in the respective direction.
- **Transition model:** Gridworld is stochastic because the actions can be unreliable. In this model, action "X" (X can be Up, Down, Left, or Right) moves you one cell in the X direction of your current position with probability 0.8, but with probabilities 0.2/3 it will transition to one of the other neighbouring cells. Transitions that take you off the grid will not result in any change.
- **Rewards:** The reward is 0 on all transitions until the terminal state is reached. Reaching terminal state gives you a reward of +10. Take the discount factor $\alpha = 0.7$.

1. Implement value iteration and policy iteration. Start with $J_0(s) = 0$ and $\pi_0(s) = \text{UP}, \forall s$. (5 marks)
2. Answer the following questions for variant with Goal 1 as terminal state: (2+2+2+2+2 marks)
   (a) Plot graph of $\max_s |J_{i+1}(s) - J_i(s)|$ vs iterations and $\sum_s \pi_{i+1}(s) \neq \pi_i(s)$ vs iterations for both value iteration and policy iteration.

(b) Compare value iteration and policy iteration by plotting $J(s)$ vs iterations for three random states. Which converges faster? Why?

(c) Show $J(s)$ and greedy policy $\pi(s)$, $\forall s$, obtained after 5 iterations, and after you stop value iteration.

(d) Show $J(s)$ and policy $\pi(s)$, $\forall s$, obtained after 5 iterations, and after you stop policy iteration.

(e) Explain the behaviour of J and greedy policy π obtained by value iteration and policy iteration.

3. Answer the following question for variant with Goal 2 as terminal state: (4 marks)

(a) Show $J(s)$ and policy $\pi(s)$, $\forall s$, obtained after you stop value iteration and policy iteration and explain it's behaviour.

Note: You have to show $J(s)$ and policy $\pi(s)$ with arrows, $\forall s$, on the image of girdworld, not in some form of table unless you explicitally make your table look like gridworld highlighting wormholes and terminal state.