CS 188
Fall 2019

Introduction to
Artificial Intelligence

Written HW 2

Due: Monday 10/14/2019 at 11:59pm (submit via Gradescope).

Policy: Can be solved in groups (acknowledge collaborators) but must be written up individually

First name	
Last name	
SID	
Collaborators	

Q1. The Game of Alice and Her Robot Bob!

Alice is playing a math game with her robot Bob. In the game, Alice and Bob take turns changing a number s. Each turn, Alice chooses an action from $\{L, M, R\}$ and Bob chooses an action randomly from $\{A, B\}$. The probability that Bob chooses action A is $0.5 \leq p < 1$. The game ends when $s = 6$, upon which Alice acquires a utility of $r = \gamma^{T-1}$, where T denotes the **total number of turns Alice took**. Figure 1 shows the transitions from s for one round of the game for different action combinations for Alice and Bob.

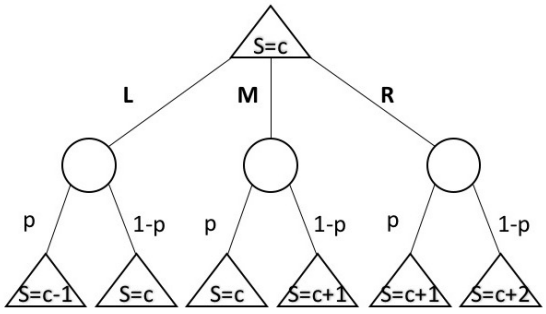


Figure 1: A depiction of the first level of the game tree. Alice can pick from L, M, R, and the robot picks A with probability p and B with probability $1 - p$. The values of s defined in each node represent the state at that node, and do **not** represent the values of those nodes. The only values come from the leaf nodes with $s = 6$ with value $r = \gamma^{T-1}$.

We provide a partially expanded game tree in Figure 2 where the triangular maximizing nodes represent Alice while the circular chance nodes represent Bob. Note that this tree has infinite depth, and the only leaves of this game tree are where $s = 6$.

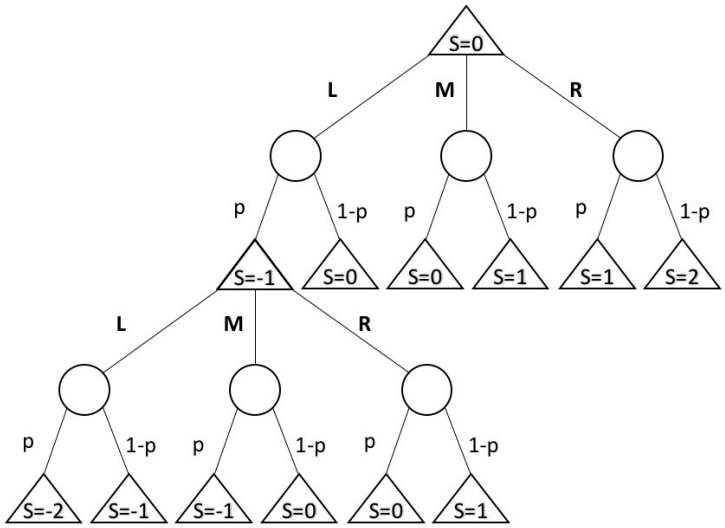


Figure 2: The first two layers of the game tree. In this game s starts at 0. Note that the second layer shows only one node expanded (there should be 6 in total).

In the question below, we tackle this problem from the perspective of an MDP.

(a) Formulate this problem into an MDP (from Alice's perspective). **Do not** include T (the number steps Alice took) in your state space. *Hint: How do the Bellman equations relate to expectimax?*

State:

Actions:

Transition Function:

Reward Function:

Discount Factor:

(b) Let $\gamma = 1$. You may write your answers in terms of p and k (introduced below). **Recall that we defined** $0.5 \leq p < 1$.

Let the policy π be such that it chooses "R" when $s < 6$ and "L" when $s > 6$.

(i) What is the expected future reward if we follow policy π starting at $s = 5$ and go for 2 steps?

(ii) What is the expected future reward if we follow policy π starting at $s = 5$ and go for k steps?

(iii) Does there exist another policy π' such that we can get a larger expected future reward if we start from $s = 5$ and act for k steps? Provide π' if yes.

Yes ◯ No ◯

If Yes, define π':

(iv) What is $V_k(s = 5)$ (the k-th iteration for $s = 5$ in the value iteration process)? Remember $V_k(s)$ can be interpreted as the largest expected reward you can get within k steps starting from s. *Hint: How does this relate to your answer from (ii)?*

(v) What are $V_k(s=4)$, $V_k(s=7)$?

(vi) What is the converged value function $V^*(s)$ for all states s?

(vii) What policy does the policy extracted from value iteration converge to (you may describe it using words or variables)?

(viii) Is this optimal policy the only optimal policy to this MDP?

Yes ◯ No ◯

If No, provide another optimal deterministic policy:

(c) Now suppose we have $0 < \gamma < 1$ and $0 < p < 1$.

(i) What is $V^*(s=7)$? You may write your answer in terms of γ, p.
Hint: Consider what the optimal policy is, and use the Bellman equations to solve for $V^*(s=7)$.

(ii) What is the value of p if $Q^*(s=5, a=M) = Q^*(s=5, a=R)$? You can include γ in your answer if you wish. *Hint: Find* $V^*(s=5), V^*(s=6)$, *and consider that* $V^*(5) = Q^*(s=5, a=M) = Q^*(s=5, a=R)$

Q2. Reward Shaping

Imagine we are teaching an agent to ride a bike to Berkeley Bowl. We could just give the agent a reward of 1 when it reaches the grocery store, but that might provide too little reward signal. We could choose to reward it at every step by the negative distance to the store, or we could reward it for forward progress. If we trained three agents with these three different reward functions we might notice something interesting.

The first agent eventually figures out how to pedal and makes its way to buy groceries. The second agent also learns to get to the store, but figures out how to do so much more quickly. However, we then realize that we forgot to penalize the third agent for backwards progress, because it starts to ride its bike in circles, accumulating infinite reward, since half the time it's getting closer to the store!

The process of modifying your reward function to make learning easier is called **reward shaping**. Practically, we would like to shape our rewards so that learning happens quickly, but that the agent still achieves the optimal behavior we have in mind. Our experiment raises a natural question: how much flexibility do we have in choosing reward functions so that the optimal behavior remains the same?

Let's investigate this formally. We define a Markov Decision Process by:

1. A state space, S.
2. A set of possible actions, A.
3. The transition dynamics, $T(s, a, s')$.
4. A discount factor $0 \leq \gamma \leq 1$.
5. A reward function $R(s, a, s')$.

The MDP is all of these things put together: $M = (S, A, T, \gamma, R)$. We're interesting in modifying the original reward function with a "reward shaping function" $F(s, a, s')$ so that the new reward function is $\hat{R}(s, a, s') = R(s, a, s') + F(s, a, s')$.

We hypothesize that to avoid altering the behavior of the optimal policy we need avoid infinite cycles of reward, like riding in circles in the case of the biker, or dumping and re-vacuuming dust in the case of the devious vacuum agent mentioned in class.

Therefore we define a **potential function** of state $\Phi(s)$, and a reward shaping function of the form $F(s, a, s') = \gamma\Phi(s') - \Phi(s)$. Intuitively, a useful potential function should output higher values for states closer to the goal, and lower values for states further away.

(a) Show that the sum of discounted additional rewards along a cyclic path $s_0 \to s_1 \to \cdots s_T \to s_0$ is 0. That is, show that

$$F(s_0, a_0, s_1) + \gamma F(s_1, a_1, s_2) + \cdots + \gamma^T F(s_T, a_T, s_0) = 0$$

Without loss of generality, let $\Phi(s_0) = 0$, since otherwise we could just choose $\hat{\Phi}(s) = \Phi(s) - \Phi(s_0)$.

(b) We will prove that the optimal policy with respect to the modified reward function remains unchanged. Recall that the Bellman Equation for state, action values under the MDP M is

$$Q^*_M(s,a) = \mathbb{E}\left[R(s,a,s') + \gamma \max_{a'} Q^*_M(s',a')\right] = \sum_{s'} T(s,a,s')\left[R(s,a,s') + \gamma \max_{a'} Q^*_M(s',a')\right]$$

and that the optimal policy is

$$\pi^*_M(s) = \arg\max_a Q^*(s,a)$$

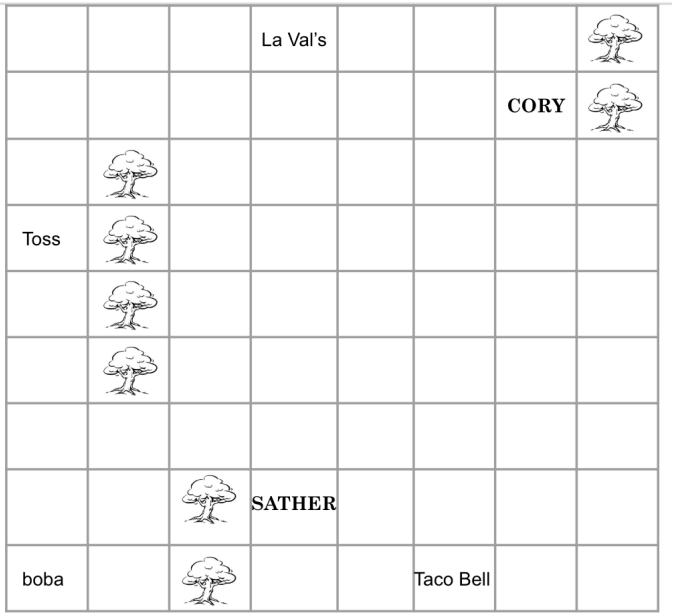
(i) Show that the optimal Q-function under the modified MDP $\hat{M} = (S, A, T, \gamma, \hat{R} = R + F)$ is $Q^*_{\hat{M}}(s,a) = Q^*_M(s,a) - \Phi(s)$. *Hint: It is sufficient to show that $Q^*_M(s,a) - \Phi(s)$ satisfies the appropriate Bellman equation for $\hat{M}$. This question is similar to Section 5 Q2(b)*

(ii) Argue that the optimal policy under the new MDP is the same as the optimal policy under the old MDP, i.e. that $\pi^*_M(s) = \pi^*_{\hat{M}}(s)$.

Let's apply reward shaping to a real world problem. Suppose that a Kiwibot food delivery robot travels in the gridworld shown below, but Kiwibot's "mapping system" no longer functions and it doesn't know how Berkeley is laid out, so it has to figure out how to navigate itself.

From any grid location without a tree, the Kiwibot is able to take an action in North, East, West, South directions, as long as the next square is valid (still on the grid and not a tree). However, whenever the Kiwibot tries to an an action, it may get knocked over with probability 0.3. An employee or good Samaritan is required to stand the Kiwibot upright in its original spot and the Kiwibot does not move that turn. Otherwise, the Kiwibot successfully moves to the next grid spot with probability 0.7.

The Kiwibot always begins in the grid marked **SATHER**.



For part (c), we model the gridworld as an MDP with state grid location, discount factor $\gamma = 1$, and actions and transitions as described in the previous paragraph. Kiwibot's goal is to reach **Cory**, and receives a reward of 1 for reaching **Cory** and receives no reward anywhere else. Once Kiwibot reaches **Cory**, its task is finished and we terminate.

(c) **(i)** We need to first define a potential function. One possibility is a function of the estimated number of steps to the goal. Assuming **no trees** exist, what is the expected time it would take an optimal policy to reach **Cory** from some state s? You may write your answer in terms of $d(s)$ = maze distance from s to **Cory**. *Hint: Calculate the expected time it takes for Kiwibot to move a single step forward, and by linearity of expectation, generalize to $d(s)$ steps. You may want to refresh on Geometric random variables.*

(ii) Note that we would want states *closer* to the goal to have *higher* potential. Thus we could define our potential function as $\Phi(s) = -t_s$, where t_s is your answer from the previous part. What is $F(s, a, s')$? You may want to refer to the definition of $F(s, a, s')$ in part (a). **Expand out your answer in terms of $d(s)$**

(iii) Suppose we run value iteration on the original MDP with $R(s, a, s')$ (only reward of 1 when reaching **Cory**). For the grid shown, what is the minimum k such that the policy extracted from $V_k(s)$ is an **optimal policy**? *Note: Remember that policy extraction does a one-step look-ahead*

(iv) Now, we run value iteration on the MDP with $R'(s, a, s') = R(s, a, s') + F(s, a, s')$, where $F(s, a, s')$ is your answer from (ii). For the grid shown, what is the minimum k such that the policy extracted from $V_k(s)$ is an **optimal policy**?

Now, let's take into account the restaurants. Suppose that the Kiwibot now needs to travels to each of the 4 restaurants (**Taco Bell**, **Boba**, **Toss**, **La Val's**) and deliver the food to **Cory**. In addition to keeping track of location, we augment the state space to include the number of locations it's visited thus far, n_s. The actions, transition probabilities, and discount $\gamma = 1$ remain unchanged from the previous question, and the Kiwibot once again only receives a reward of 1 when it reaches **Cory**.

(d) **(i)** Suppose that the expected number of steps from any location to the next is always about t steps (this is an estimate). Let our potential function be $\Phi(s) = -\frac{5-n_s}{5}t$. Intuitively, this assigns a higher potential for states that have visited more restaurants, and approximates the future cost to the goal using the number of places to visit left, each with t steps required in-between each location. What is $F(s, a, s')$?

(ii) When we reach a new restaurant, how much does our modified reward function $R'(s, a, s')$ increase with $F(s, a, s')$ accounted for (in terms of t)?

(iii) When is $F(s, a, s')$ non-zero (you may describe this using words)? How does this fall in-line with our intuition for how we might do manual reward-shaping?