CS 234 Winter 2019
Assignment 1
Due: January 23 at 11:59 pm

For submission instructions please refer to website

1 Optimal Policy for Simple MDP [20 pts]

Consider the simple n-state MDP shown in Figure 1. Starting from state s_1, the agent can move to the right (a_0) or left (a_1) from any state s_i. Actions are deterministic and always succeed (e.g. going left from state s_2 goes to state s_1, and going left from state s_1 transitions to itself). Rewards are given upon taking an action from the state. Taking any action from the goal state G earns a reward of $r = +1$ and the agent stays in state G. Otherwise, each move has zero reward ($r = 0$). Assume a discount factor $\gamma < 1$.

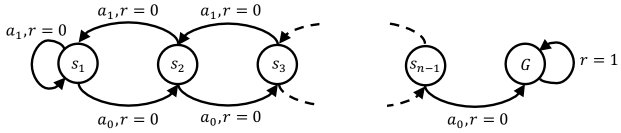

Figure 1: n-state MDP

(a) The optimal action from any state s_i is taking a_0 (right) until the agent reaches the goal state G. Find the optimal value function for all states s_i and the goal state G. [5 pts]

solution:

$$V(G) = \frac{1}{1-\gamma}$$
$$V(s_{n-1}) = \frac{\gamma}{1-\gamma}$$
$$\vdots$$
$$V(s_1) = \frac{\gamma^{n-1}}{1-\gamma}$$

(b) Does the optimal policy depend on the value of the discount factor γ? Explain your answer. [5 pts]

solution: If $\gamma > 0$, value of γ does not change the ordering of states, so the optimal policy is the same; however, the value of the value function depends on γ. If $\gamma = 0$ then, policy $\forall s : \pi(s) = a_0$ is still an optimal policy; however, this is not the only optimal policy.

(c) Consider adding a constant c to all rewards (i.e. taking any action from states s_i has reward c and any action from the goal state G has reward $1 + c$). Find the new optimal value function for all states s_i and the goal state G. Does adding a constant reward c change the optimal policy? Explain your answer. [5 pts]

solution: No effect on the optimal policy. Adding a constant c to all the rewards only changes the value of each state by a constant v_c for any policy π:

$$\begin{aligned} v^{\pi}_{new}(s_i) &= \sum_{t=0}^{\infty} \gamma^t (r_t + c) \\ &= \sum_{t=0}^{\infty} \gamma^t r_t + \sum_{t=0}^{\infty} \gamma^t c \\ &= v^{\pi}_{old}(s_i) + \frac{c}{1-\gamma} \end{aligned}$$

(d) After adding a constant c to all rewards now consider scaling all the rewards by a constat a (i.e. $r_{new} = a(c + r_{old})$). Find the new optimal value function for all states s_i and the goal state G. Does that change the optimal policy? Explain your answer, If yes, give an example of a and c that changes the optimal policy. [5 pts]

solution:

$$\begin{aligned} v^{\pi}_{new}(s_i) &= \sum_{t=0}^{\infty} \gamma^t a(r_t + c) \\ &= a\sum_{t=0}^{\infty} \gamma^t r_t + \sum_{t=0}^{\infty} \gamma^t ac \\ &= av^{\pi}_{old}(s_i) + \frac{ac}{1-\gamma} \end{aligned}$$

So if $a > 0$ then the optimal policy will not change, and the value of the new optimal policy is a linear mapping of the previous optimal value function $av^*(s_i) + \frac{ac}{1-\gamma}$.

If $a = 0$ then all states have reward 0 and any policy is the optimal policy, and the optimal value of all states is 0.

If $a < 0$, any policy that never reaches to the state G is the optimal policy with value $\frac{ac}{1-\gamma}$ for all states s_i and $\frac{a(1+c)}{1-\gamma}$ for state G.

2 Running Time of Value Iteration [20 pts]

In this problem we construct an example to bound the number of steps it will take to find the optimal policy using value iteration. Consider the infinite MDP with discount factor $\gamma < 1$ illustrated in Figure 2. It consists of 3 states, and rewards are given upon taking an action from the state. From state s_0, action a_1 has zero immediate reward and causes a deterministic transition to state s_1 where there is reward +1 for every time step afterwards (regardless of action). From state s_0, action a_2 causes a deterministic transition to state s_2 with immediate reward of $\gamma^2/(1-\gamma)$ but state s_2 has zero reward for every time step afterwards (regardless of action).

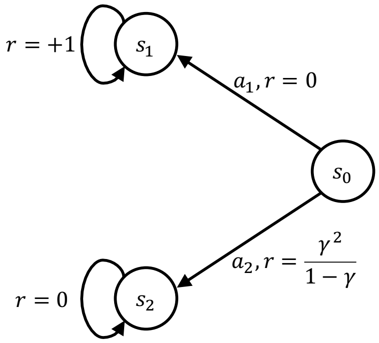

Figure 2: infinite 3-state MDP

(a) What is the total discounted return ($\sum_{t=0}^{\infty} \gamma^t r_t$) of taking action a_1 from state s_0 at time step $t = 0$? [5 pts]

solution: $V = 0 + \gamma + \gamma^2 + \dots = \frac{\gamma}{1-\gamma}$

(b) What is the total discounted return ($\sum_{t=0}^{\infty} \gamma^t r_t$) of taking action a_2 from state s_0 at time step $t = 0$? What is the optimal action? [5 pts]

solution: $V = \frac{\gamma^2}{1-\gamma} + 0 + 0 + \dots = \frac{\gamma^2}{1-\gamma}$, so the optimal action is a_1

(c) Assume we initialize value of each state to zero, (i.e. at iteration $n = 0$, $\forall s : V_{n=0}(s) = 0$). Show that value iteration continues to choose the sub-optimal action until iteration n^* where,

$$n^* \geq \frac{\log(1-\gamma)}{\log \gamma} \geq \frac{1}{2} \log(\frac{1}{1-\gamma}) \frac{1}{1-\gamma}$$

Thus, value iteration has a running time that grows faster than $1/(1-\gamma)$. (You just need to show the first inequality) [10 pts]

solution: For all iterations $V_n(s_2) = 0$, so $Q(s_0, a_2) = \frac{\gamma^2}{1-\gamma}$. Value iteration keep choosing the sub-optimal action while $Q(s_0, a_2) > Q(s_0, a_1)$. Value iteration updates are as follows:

$$\begin{aligned} Q_{n+1}(s_0, a_1) &= 0 + \gamma V_n(s_1) \\ V_{n+1}(s_1) &= 1 + \gamma V_n(s_1) \end{aligned}$$

So,

$$\begin{aligned} Q_{n+1}(s_0, a_1) &= 0 + \gamma(1 + \gamma V_n(s_1)) \\ &= \gamma(1 + \gamma + \dots + \gamma^{n-1} + \gamma^n V_{n=0}(s_1)) \\ &= \gamma(\frac{1-\gamma^n}{1-\gamma}) \end{aligned}$$

Setting this equal to $Q(s_0, a_2)$:

$$\begin{aligned}
\gamma(\frac{1-\gamma^{n^*}}{1-\gamma}) &= \frac{\gamma^2}{1-\gamma} \\
n^* &= \frac{\log 1-\gamma)}{log(\gamma)} \\
&= \frac{\log(1-\gamma)}{log(1+\gamma-1)} \\
&\geq \log(1-\gamma)\frac{2+\gamma-1}{2(\gamma-1)} \\
&= -\log 1/(1-\gamma)\frac{\gamma+1}{-2(1-\gamma)} \\
&\geq \frac{1}{2}\log(\frac{1}{1-\gamma})\frac{1}{1-\gamma}
\end{aligned}$$

Where the first inequality follows by $\log(1+x) \leq \frac{2x}{2+x}$ for $x \in (-1, 0]$, and the log is natural logarithm.

3 Approximating the Optimal Value Function [35 pts]

Consider a finite MDP $M = \langle S, A, T, R, \gamma \rangle$, where S is the state space, A action space, T transition probabilities, R reward function and γ the discount factor. Define Q^* to be the optimal state-action value $Q^*(s, a) = Q_{\pi^*}(s, a)$ where π^* is the optimal policy. Assume we have an estimate $\tilde{Q}$ of Q^*, and $\tilde{Q}$ is bounded by l_∞ norm as follows:

$$||\tilde{Q} - Q^*||_\infty \leq \varepsilon$$

Where $||x||_\infty = max_{s,a}|x(s, a)|$.

Assume that we are following the greedy policy with respect to $\tilde{Q}$, $\pi(s) = argmax_{a \in \mathcal{A}}\tilde{Q}(s, a)$. We want to show that the following holds:

$$V_\pi(s) \geq V^*(s) - \frac{2\varepsilon}{1-\gamma}$$

Where $V_\pi(s)$ is the value function of the greedy policy π and $V^*(s) = max_{a \in A}Q^*(s, a)$ is the optimal value function. This shows that if we compute an approximately optimal state-action value function and then extract the greedy policy for that approximate state-action value function, the resulting policy still does well in the real MDP.

(a) Let π^* be the optimal policy, V^* the optimal value function and as defined above $\pi(s) = argmax_{a \in A}\tilde{Q}(s, a)$. Show the following bound holds for all states $s \in S$. [10 pts]

$$V^*(s) - Q^*(s, \pi(s)) \leq 2\varepsilon$$

solution: By construction of π, $\tilde{Q}(s, \pi(s)) \geq \tilde{Q}(s, \pi^*(s))$.

$$\begin{aligned} V^*(s) - Q^*(s, \pi(s)) &= V^*(s) - \tilde{Q}(s, \pi(s)) + \tilde{Q}(s, \pi(s)) - Q^*(s, \pi(s)) \\ &\leq V^*(s) - \tilde{Q}(s, \pi^*(s)) + \varepsilon \\ &= Q^*(s, \pi^*(s)) - \tilde{Q}(s, \pi^*(s)) + \varepsilon \\ &\leq 2\varepsilon \end{aligned}$$

(b) Using the results of part 1, prove that $V_\pi(s) \geq V^*(s) - \frac{2\varepsilon}{1-\gamma}$. [10 pts]

solution:

$$\begin{aligned} V^*(s) - V_\pi(s) &= V^*(s) - Q^*(s, \pi(s)) + Q^*(s, \pi(s)) - V_\pi(s) \\ &\leq 2\varepsilon + Q^*(s, \pi(s)) - Q_\pi(s, \pi(s)) \\ &= 2\varepsilon + \gamma \mathbb{E}_{s'}[V^*(s') - V_\pi(s')] \end{aligned}$$

By recursing on this equation and using linearity of expectation we get $V_\pi(s) \geq V^*(s) - \frac{2\varepsilon}{1-\gamma}$.

Now we show that this bound is tight. Consider the 2-state MDP illustrated in figure 3. State s_1 has two actions, "*stay*" self transition with reward 0 and "*go*" that goes to state s_2 with reward 2ε. State s_2 transitions to itself with reward 2ε for every time step afterwards.

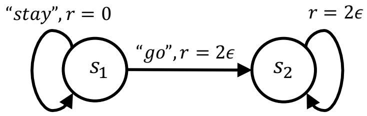

Figure 3: 2-state MDP

(c) Compute the optimal value fucntion $V^*(s)$ for each state and the optimal state-action value function $Q^*(s, a)$ for state s_1 and each action. [5 pts]

solution:

$$\begin{aligned} Q^*(s_1, go) &= \frac{2\varepsilon}{1-\gamma} \\ Q^*(s_1, stay) &= \frac{2\varepsilon\gamma}{1-\gamma} \\ V^*(s_1) &= \frac{2\varepsilon}{1-\gamma} \\ V^*(s_2) &= \frac{2\varepsilon}{1-\gamma} \end{aligned}$$

(d) Show that there exists an approximate state-action value function $\tilde{Q}$ with ε error (measured with l_∞ norm), such that $V_\pi(s_1) - V^*(s_1) = -\frac{2\varepsilon}{1-\gamma}$, where $\pi(s) = argmax_{a \in A} \tilde{Q}(s, a)$. (You may need to define a consistent tie break rule) [10 pts]

solution: As observed the difference between two state-value function is 2ε, so one can simply build a state-action value function $\tilde{Q}$ that makes $\pi(s_1) = stay$ the optimal action at s_1 (set $\tilde{Q}(s_1, go) = Q^*(s_1, go) - \varepsilon$, and $\tilde{Q}(s_1, stay) = Q^*(s_1, stay) + \varepsilon$), and $V_\pi(s_1) - V^*(s_1) = -\frac{2\varepsilon}{1-\gamma}$. So the bound is tight.

4 Frozen Lake MDP [25 pts]

Now you will implement value iteration and policy iteration for the Frozen Lake environment from OpenAI Gym. We have provided custom versions of this environment in the starter code.

(a) **(coding)** Read through `vi_and_pi.py` and implement `policy_evaluation`, `policy_improvement` and `policy_iteration`. The stopping tolerance (defined as $\max_s |V_{old}(s) - V_{new}(s)|$) is tol $=$ 10^{-3} . Use $\gamma = 0.9$. Return the optimal value function and the optimal policy. [10pts]

(b) **(coding)** Implement `value_iteration` in `vi_and_pi.py`. The stopping tolerance is tol $=$ 10^{-3} . Use $\gamma = 0.9$. Return the optimal value function and the optimal policy. [10 pts]

(c) **(written)** Run both methods on the Deterministic-4x4-FrozenLake-v0 and

Stochastic-4x4-FrozenLake-v0 environments. In the second environment, the dynamics of the world are stochastic. How does stochasticity affect the number of iterations required, and the resulting policy? [5 pts]

solution: Stochasticity generally increases the number of iterations required to converge. In the stochastic frozen lake environment, the number of iterations for value iteration increases. For policy iteration, depending on the implementation method, the number of iterations could remain unchanged; or policy iteration might not even converge at all. The stochasticity would also change the optimal policy. In this environment, the optimal policy of the stochastic frozen lake is different from the one of the deterministic frozen lake.