CS 234 Winter 2018
Assignment 1
Due: January 24 at 11:00 pm

Assignment Owners: Andrea Zanette, Xinkun Nie

1 Effect of Modeling Errors in Policy Evaluation [20 pts]

Consider deploying a Reinforcement Learning (RL) agent on an episodic MDP M with a horizon of H timesteps. The true MDP M is never revealed to the agent except for the state and action space (S and A, respectively) and the time horizon H. In other words, the agent knows neither the expected reward $r(s,a)$ nor the transition dynamics $P(s' \mid s,a)$ for a generic state-action pair (s,a).

As the agent explores the environment in different episodes it builds an empirical model of the world (often via maximum likelihood) which we denote by $\hat{M}$. After collecting sufficient data we would expect the empirical MDP to be similar to the true MDP.

Since our MDP is episodic, the value of a state depends on the time to termination of the episode. Thus, for a given stochastic policy π, we require a different value function for each timestep, which we denote $V_1, \ldots, V_H$ in M and $\hat{V}_1, \ldots, \hat{V}_H$ in $\hat{M}$. In particular, for $i = 1, \ldots, H$, let

$$V_i(s) = \mathbb{E}\left[\sum_{t=i}^{H} r(s_t, a_t) \;\middle|\; s_i = s\right] \text{ and } \hat{V}_i(s) = \mathbb{E}\left[\sum_{t=i}^{H} \hat{r}(s_t, a_t) \;\middle|\; s_i = s\right],$$

where the expectation is taken under following policy π. Suppose we use the empirical MDP $\hat{M}$ instead of M to evaluate policy π. Assuming $\hat{V}_{H+1} = V_{H+1} = \vec{0}$ show that for all $i = 1, \ldots, H$:

$$\hat{V}_i(s) - V_i(s) = \sum_{t=i}^{H} \mathbb{E}\left[\hat{r}(s_t, a_t) - r(s_t, a_t) + \sum_{s'} (\hat{P}(s' \mid s_t, a_t) - P(s' \mid s_t, a_t))\hat{V}_{t+1}(s') \;\middle|\; s_i = s\right].$$

In the above equality the expectation is defined with respect to the states encountered in true MDP M upon starting from s_i and following stochastic policy π.

This result relates the value of policy π on $\hat{M}$ and M using the expected trajectories on M which we can compute easily. If it holds that $\hat{r}$ and $\hat{P}$ are close to r and P then this result can be used to conclude that the empirical value function $\hat{V}$ is also close to the true one V.

If you use an existing result to derive the result, you need to cite the source.

2 Value Iteration [35 pts]

(a) After each iteration of value iteration you can extract the current best policy (given the current value function, by taking an argmax instead of max). If the best policy at step k is the same as the best policy at step k+1 when doing value iteration, can the policy ever change again (with further iterations of value iteration)? Prove it cannot or disprove with a counterexample.

(b) Consider an MDP with finite state space and finite action set. Empirically we often halt value iteration after a fixed number of steps for computational reasons, yielding an approximately optimal value function $\tilde{V}$. We then extract a greedy policy $\pi_{\tilde{V}}$ from $\tilde{V}$ by taking the argmax of one more backup. Let's assume we know that the maximum difference across any state is (for extra understanding, think about how the difference in $|BV - V|$ allows us to ensure this)

$$|V^*(s) - \tilde{V}(s)| \leq \varepsilon, \text{for all } s.$$

Then policy $\pi_{\tilde{V}}$ has values close to the optimal ones for each state. In particular, define the loss function $L_{\tilde{V}}$ as follows:

$$L_{\tilde{V}}(s) = V^*(s) - V_{\pi_{\tilde{V}}}(s), \text{for all } s,$$

where V^* is the optimal value function and $V_{\pi_{\tilde{V}}}$ is the value function under policy $\pi_{\tilde{V}}$. Prove

$$L_{\tilde{V}}(s) \leq \frac{2\gamma\varepsilon}{1-\gamma}$$

for all states s. Here $\gamma < 1$ is the discount factor.

If you use outside sources you must write up your own solution and cite the used source.

3 Grid Policies [20 pts]

Consider the following grid environment. Starting from any unshaded square, you can move up, down, left, or right. Actions are deterministic and always succeed (e.g. going left from state 1 goes to state 0) unless they will cause the agent to run into a wall. The thicker edges indicate walls, and attempting to move in the direction of a wall results in staying in the same square. Taking any action from the green target square (no. 5) earns a reward of +5 and ends the episode. Taking any action from the red square of death (no. 11) earns a reward of -5 and ends the episode. Otherwise, each move is associated with some reward $r \in \{-1, 0, +1\}$. Assume the discount factor $\gamma = 1$ unless otherwise specified.

0	1	2	3	4
5	6	7	8	9
10	11	12	13	14
15	16	17	18	19
20	21	22	23	24

(a) Define the reward r for all states (except state 5 and state 11 whose rewards are specified above) that would cause the optimal policy to return the shortest path to the green target square (no. 5).

(b) Using r from part (a), find the optimal value function for each square.

(c) Does setting $\gamma = 0.8$ change the optimal policy? Why or why not?

(d) All transitions are even better now: each transition now has an extra reward of 1 in addition to the reward you defined in (a). Assume $\gamma = 0.8$ as in part (c). How would the value function change? How would the policy change? Explain why.

4 Frozen Lake MDP [25 pts]

Now you will implement value iteration and policy iteration for the Frozen Lake environment from OpenAI Gym (https://gym.openai.com/envs/FrozenLake-v0). We have provided custom versions of this environment in the starter code.

(a) **(coding)** Implement policy iteration in `vi_and_pi.py`. 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]