CS 277 (W22): Control and Reinforcement Learning
Assignment 4

Due date: Friday, March 4, 2022 (Pacific Time)

Roy Fox
https://royf.org/crs/W22/CS277

Instructions: In theory questions, a formal proof is not needed (unless specified otherwise). instead, briefly explain informally the reasoning behind your answers.

In practice questions, include a printout of your code as a page in your PDF, and a screenshot of TensorBoard learning curves (`episode_reward_mean`, unless specified otherwise) as another page.

Part 1 Model-based error accumulation (25 points + 5 bonus)

Consider a model-based reinforcement learning algorithm that estimates a model $\hat{p}$ of the true dynamics p, and then uses it for planning. In all parts of this question, we assume that we can plan optimally in the estimated model, with the true non-negative reward function.

Question 1 (10 points + 5 bonus) Suppose that the estimated model is guaranteed, for some $\epsilon > 0$, to be an ϵ-approximation, i.e. have

$$\|p(s'|s,a) - \hat{p}(s'|s,a)\|_1 \leq \epsilon,$$

for all s and a, and that the initial distribution $p(s_0)$ is known exactly. Show that, for any policy π

$$\mathbb{E}_{(s_t,a_t)\sim p_\pi}[r(s_t,a_t)] - \mathbb{E}_{(s_t,a_t)\sim \hat{p}_\pi}[r(s_t,a_t)] \leq \epsilon t r_{\max}.$$

Hint: show by induction that, for any $t \geq 0$, and state s $\|p_\pi(s_t = s) - \hat{p}_\pi(s_t = s)\|_1 \leq \epsilon t$.

Bonus: show the tighter bound

$$\mathbb{E}_{(s_t,a_t)\sim p_\pi}[r(s_t,a_t)] - \mathbb{E}_{(s_t,a_t)\sim \hat{p}_\pi}[r(s_t,a_t)] \leq \tfrac{1}{2}\epsilon t r_{\max}.$$

Question 2 (5 points) Conclude that planning with $\hat{p}$ is near-optimal: if π is optimal for p and $\hat{\pi}$ is optimal for $\hat{p}$, for discount factor γ, then

$$\mathbb{E}_{\xi\sim p_\pi}[R(\xi)] - \mathbb{E}_{\xi\sim p_{\hat{\pi}}}[R(\xi)] \leq 2\tfrac{\gamma}{(1-\gamma)^2}\epsilon r_{\max}.$$

Or, given the bonus question above, halve the RHS.

Hint: recall that $\sum_t \gamma^t t = \frac{\gamma}{(1-\gamma)^2}$.

Question 3 (10 points) Now suppose instead that the state space is $\mathbb{R}^n$, and that both the true dynamics $f : \mathbb{R}^n \to \mathbb{R}^n$ and the model $\hat{f} : \mathbb{R}^n \to \mathbb{R}^n$ are deterministic, with a known initial state s_0. Determinism implies that there exists an optimal open-loop policy, i.e. a sequence of actions.

Suppose that the true dynamics, the model, and the reward function are all Lipschitz. That is, there exists a real constant L such that, for all states s and $\hat{s}$ and action a

$$\|f(s,a) - f(\hat{s},a)\|_2 \leq L\|s - \hat{s}\|_2,$$

and similarly for $\hat{f}$ and for r, i.e. $|r(s,a) - r(\hat{s},a)| \leq L\|s - \hat{s}\|_2$. Suppose further that the estimated model is guaranteed, for some $\epsilon > 0$, to be an ϵ-approximation, i.e have

$$\|f(s,a) - \hat{f}(s,a)\|_2 \leq \epsilon,$$

for all s and a.

Fix an action sequence $\vec{a} = a_0, a_1, \ldots$. Denote the resulting state sequence when rolling out $\vec{a}$ in f by $s_0, s_1, \ldots$, and in $\hat{f}$ by $\hat{s}_0, \hat{s}_1, \ldots$ (note that $s_0 = \hat{s}_0$). Show by induction that, for any $t \geq 0$

$$|r(s_t, a_t) - r(\hat{s}_t, a_t)| \leq \frac{L^t - 1}{L - 1} L\epsilon,$$

assuming $L \neq 1$.

Part 2 Finite-state controllers (25 points)

A finite-state controller (FSC) π is a finite-state machine with: (1) a finite set $\mathcal{M}$ of memory states; (2) an memory state update distribution $\pi(m_t|m_{t-1}, o_t)$, giving the probability of updating from internal state m_{t-1}, upon observing o_t, to m_t; and (3) an action distribution $\pi(a_t|m_t)$.

Question 1 (10 points) Given a POMDP with dynamics $p(s_{t+1}|s_t, a_t)$ and observation model $p(o_t|s_t)$, and an FSC π, write down a forward recursion for computing the joint distribution of m_{t-1} and s_t. That is, show how to compute $p_\pi(m_t, s_{t+1})$ using p, π, and $p_\pi(m_{t-1}, s_t)$.

Question 2 (5 points) Given the joint distribution of (m_{t-1}, s_t), show how to compute the Bayesian predictive belief $b' = p(s_t|m_{t-1})$.

Question 3 (10 points) Given also a reward function $r(s_t, a_t)$, write down a backward recursion for evaluating $V_\pi(s_t, m_t)$. That is, show how to compute $V_\pi(s_t, m_t)$ using p, π, r, and $V_\pi(s_{t+1}, m_{t+1})$.

Part 3 RNN policies (50 points)

Question 1 (15 points) In the `LunarLander` environment (`https://gym.openai.com/envs/LunarLander-v2/`), the observation is:

$$[x \text{ position}, y \text{ position}, x \text{ velocity}, y \text{ velocity}, \text{orientation}, \text{angular velocity},$$
$$\text{left leg contact (Boolean)}, \text{right leg contact (Boolean)}].$$

In the `Pong` environment (https://gym.openai.com/envs/Pong-v0/), the observation is the image that the Atari console would render to the screen (usually 84×84 grayscale pixels, after cropping, rescaling, and gray-scaling) . Alternatively, Atari environments are often "wrapped" to provide in every step the 4 most recent images, i.e. an observation shaped $4 \times 84 \times 84$ (this is called *frame-stacking*).

In which of these 3 environments (`LunarLander`, `Pong`, and frame-stacked `Pong`) would you expect an agent to benefit the most and the least from having memory, compared with a memoryless policy?

Question 2 (35 points) Test your hypothesis. Use any algorithm implemented in RLlib (https://docs.ray.io/en/latest/rllib-toc.html#algorithms) with a memoryless policy, and with an RNN policy (by setting `use_lstm` to `True`). Report your findings.