CS 277 (W26): Control and Reinforcement Learning
Exercise 4

Due date: Tuesday, March 3, 2026 (Pacific Time)

Roy Fox
https://royf.org/crs/CS277/W26

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 Properties of linear–Gaussian systems (20 points)

Question 1.1 (7 points) Consider a deterministic uncontrolled LTI system with dynamics $x_{t+1} = Ax_t$, where A is an $n \times n$ transition matrix, that is only observable through a noiseless partial observation $y_t = Cx_t$, where C is a $k \times n$ observation matrix. The *observability matrix* of the system (A, C) is

$$O = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}.$$

We say that a state $x \neq 0$ is *unobservable* if, after starting at $x_0 = x$, we have only zero observations, i.e. $y_t = 0$ for all $t \geq 0$. Show that there exists an unobservable state $x \neq 0$ if and only if the rank of O is less than n. Hint: by the rank–nullity theorem, the rank and the dimension of the kernel ($\ker O = \{x \mid Ox = 0\}$) sum to the dimension of the domain, in this case n.

Question 1.2 (7 points) A system (A, C) as in the previous question whose observability matrix has full column rank (i.e. rank n) is called *fully observable*. Show that a system is fully observable if and only if we can uniquely find what x_0 was at time 0 after seeing enough observations $y_0, \ldots, y_{t-1}$. Guidance: in one direction, use the fact that any full column-rank matrix M has a left inverse $M^\dagger M = I$. In the other direction, show that if $x_0 = x$ and $x_0 = x'$ induce the same observation sequence, then there exists an unobservable state.

Question 1.3 (6 points) When A itself isn't full-rank, i.e. it maps some states to 0, some information about x_0 may be lost by the dynamics and never become observable. On the other hand, only the current state x_t matters for control and future costs, so we may not actually need that information anyway. Show that, if $\ker O \subseteq \ker A^n$, then we can uniquely find x_n from the observations $y_0, \ldots, y_{n-1}$.[1]

[1] As an aside, the other direction is also true, but you don't need to show it.

Hint: show that, under the question's assumptions, if $x_0 = x$ and $x_0 = x'$ induce the same $y_0, \ldots, y_{n-1}$, then they also induce the same x_n.

Part 2 Model-based error accumulation (30 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, assume that we can plan optimally in the estimated model with the true reward function r, and that rewards are non-negative and bounded by $r_{\max}$.

Question 2.1 (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 and time t

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

That is, that π does at most $\epsilon t r_{\max}$ worse on $\hat{p}$ than on p.
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 2.2 (10 points) Conclude that planning optimally in $\hat{p}$ is also near-optimal in p: 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 term on the right-hand side.
Hint: note that the expression above only involves p_π and $p_{\hat{\pi}}$; how does that relate to $\hat{p}$?
Another hint: recall that $\sum_t \gamma^t t = \frac{\gamma}{(1-\gamma)^2}$.

Question 2.3 (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 $L \geq 0$ such that, for all states s and $\hat{s}$ and action a

$$\begin{aligned}\|f(s,a) - f(\hat{s},a)\|_2 &\leq L\|s - \hat{s}\|_2\\ \|\hat{f}(s,a) - \hat{f}(\hat{s},a)\|_2 &\leq L\|s - \hat{s}\|_2\\ |r(s,a) - r(\hat{s},a)| &\leq L\|s - \hat{s}\|_2.\end{aligned}$$

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 3 RNN policies (50 points)

Note: It takes about 10 hours to complete one Atari run on a GPU. Please keep the following in mind:

1. When submitting a Slurm job, request *more than* 10 hours of wall time (e.g., 11–12 hours). Otherwise, the job may terminate mid-training when the time limit is reached.
2. This exercise requires four Atari runs and two Acrobot runs. You can run them in parallel by submitting multiple Slurm jobs. If you are using a single machine, start early so you do not fall behind.

Question 3.1 (15 points) In the `Acrobot` environment (https://gymnasium.farama.org/environments/classic_control/acrobot), the observation is:

$$[\cos\theta_1, \sin\theta_1, \cos\theta_2, \sin\theta_2, \text{angular velocity of } \theta_1, \text{angular velocity of } \theta_2]$$

where θ_1 and θ_2 are angles of the first and second joints. In the `Seaquest` environment (https://ale.farama.org/environments/seaquest/), 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 three settings (`Acrobot`, `Seaquest`, and frame-stacked `Seaquest`) would you expect an agent to benefit the *most* and the *least* from adding recurrence (an RNN policy) relative to a memoryless policy? Briefly justify your answer.

Question 3.2 (35 points) Test your hypothesis. Use PPO implemented in Stable-Baselines3 (https://stable-baselines3.readthedocs.io/en/master/modules/ppo.html) with a memoryless policy and with an RNN policy (by setting `algo` to `ppo_lstm`). Attach TensorBoard graphs and report your findings.
For example, `submit_ppolstm_seaquestFS_gpu.sh` in https://royf.org/crs/CS277/W26/CS277E4.zip uses PPO with an LSTM policy for `Seaquest` with frame stacking.
Please note that you should have `rl_zoo3` and `atari` installed:

```
pip install rl_zoo3
pip install gymnasium[atari]
```