ECE586IC: Interplay between Control and Machine Learning **Fall 2026**

Homework 3

Instructor: Bin Hu Due date: December 1, 2026

1. (20 points) **Probability and MDPs**

(a) Let $B_1, B_2, \ldots, B_m$ be events such that exactly one of them must occur (they are mutually exclusive, and together they cover all possible outcomes). Suppose $P(B_i) > 0$ for all i. Prove the law of total probability: for any event E,

$$P(E) = \sum_{i=1}^{m} P(E \,|\, B_i)\, P(B_i).$$

You may use $P(E \,|\, B) = P(E \cap B)/P(B)$.

(b) Consider a finite MDP $\langle \mathcal{S}, \mathcal{A}, P, c, \gamma \rangle$ with state space $\mathcal{S} = \{1, 2, \ldots, n\}$ and action space $\mathcal{A} = \{a_1, \ldots, a_r\}$. Define $P_{ij}^a := P(s_{k+1} = j \,|\, s_k = i,\, a_k = a)$. The stage cost is $c(i, a)$, and $0 < \gamma < 1$ is the discount factor. The goal is to find a policy that minimizes $\mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k c(s_k, a_k)\right]$. Let π be a stochastic policy with $\pi(a \mid i) \geq 0$ and $\sum_{a \in \mathcal{A}} \pi(a \mid i) = 1$ for each state i. Under π, $\{s_k\}$ becomes a Markov chain. Use the law of total probability to derive its transition matrix P^π and find the expected stage cost $c^\pi(i) = \mathbb{E}[c(s_k, a_k) \,|\, s_k = i]$. In addition, define $V^\pi(i) = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k c(s_k, a_k) \,\middle|\, s_0 = i\right]$, and let $\mathcal{Q}^\pi(i, a)$ be the same expectation with $a_0 = a$ and π followed afterwards. Derive the Bellman equations for V^π and $\mathcal{Q}^\pi$. Show how to compute V^π by solving a linear system, and then obtain $\mathcal{Q}^\pi$ from V^π.

(c) Consider the MDP in (b) and specialize to $\mathcal{A} = \{a_1, a_2\}$ with $\pi(a_1 \mid i) = p_1$ and $\pi(a_2 \mid i) = 1 - p_1$ in every state, where $0 < p_1 < 1$. Suppose the transition model is unknown, but we observe a long trajectory $\{(s_k, a_k, c_k)\}$ generated by π, where $c_k = c(s_k, a_k)$. Write the update rule of TD(0) for estimating V^π from this trajectory.

(d) Suppose we use π as the behavior policy to generate the trajectory for the two-action MDP in (c). Write out the $\mathcal{Q}$-learning update rule. What is the size of the $\mathcal{Q}$ table? In addition, write out the SARSA update rule. Explain the difference between SARSA and $\mathcal{Q}$-learning.

2. (20 points) **Linear Quadratic Regulator**
Consider the linear time-invariant system

$$x_{k+1} = Ax_k + Bu_k + w_k$$

where x_k is the state, u_k is the control action, and the process noise w_k is sampled from a Gaussian distribution in an IID manner, i.e. $w_k \sim \mathcal{N}(0, W)$. The objective is to choose u_k to minimize the following cost

$$\mathcal{C} = \sum_{k=0}^{\infty} \gamma^k \mathbb{E}(x_k^\mathsf{T} Q x_k + u_k^\mathsf{T} R u_k)$$

where $0 < \gamma < 1$ is the discount factor. The matrices Q and R are positive definite. To avoid confusion with the weight matrix Q, we use $\mathcal{Q}$ to denote state-action value functions.

(a) Policy evaluation: Consider a linear policy $u_k = -Kx_k$, where K is stabilizing in the sense that the spectral radius of $A - BK$ is smaller than $1/\sqrt{\gamma}$ (this guarantees that the cost is finite). Define the state value function and the state-action value function as

$$\mathcal{C}_K(x) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k (x_k^\mathsf{T} Q x_k + u_k^\mathsf{T} R u_k) \,\middle|\, x_0 = x\right],$$

$$\mathcal{Q}_K(x,u) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k (x_k^\mathsf{T} Q x_k + u_k^\mathsf{T} R u_k) \,\middle|\, x_0 = x,\, u_0 = u\right],$$

where $u_k = -Kx_k$ for all $k \geq 1$ in the definition of $\mathcal{Q}_K$. Derive the Bellman equations for $\mathcal{C}_K$ and $\mathcal{Q}_K$. Then show that

$$\mathcal{C}_K(x) = x^\mathsf{T} P_K x + q_K, \qquad \mathcal{Q}_K(x,u) = \begin{bmatrix} x \\ u \end{bmatrix}^\mathsf{T} H_K \begin{bmatrix} x \\ u \end{bmatrix} + q_K,$$

where P_K is the solution of the Lyapunov equation

$$P_K = Q + K^\mathsf{T} R K + \gamma (A - BK)^\mathsf{T} P_K (A - BK).$$

Find the matrix H_K and the constant q_K in terms of P_K.

(b) Approximate policy iteration: By (a), $\mathcal{Q}_K(x,u) = \phi(x,u)^\mathsf{T}\theta_K$ for some parameter vector θ_K, where the feature vector $\phi(x,u)$ consists of the constant 1 and all quadratic monomials $z_i z_j$ $(i \leq j)$ of $z = \begin{bmatrix} x^\mathsf{T} & u^\mathsf{T} \end{bmatrix}^\mathsf{T}$. Suppose the model is unknown, but we observe transitions (x_k, u_k, c_k, x_{k+1}) for $k = 1, \ldots, N$, where $c_k = x_k^\mathsf{T} Q x_k + u_k^\mathsf{T} R u_k$.

- Use the Bellman equation for $\mathcal{Q}_K$ to derive the least-squares temporal difference (LSTD-Q) estimate of θ_K from these N transitions. (Hint: Define the temporal difference error $\delta_k(\theta) = c_k + \gamma \phi(x_{k+1}, -Kx_{k+1})^\mathsf{T}\theta - \phi(x_k, u_k)^\mathsf{T}\theta$, where the next action is $-Kx_{k+1}$ rather than u_{k+1}. LSTD-Q chooses θ such that the TD errors are orthogonal to the features, i.e. $\sum_{k=1}^{N} \phi(x_k, u_k)\, \delta_k(\theta) = 0$. This is different from minimizing $\sum_{k=1}^{N} \delta_k(\theta)^2$.)

- Explain why the data should be generated with exploration noise, e.g. $u_k = -Kx_k + e_k$ with e_k being IID zero-mean noise, rather than with $u_k = -Kx_k$. (Hint: what happens to the feature vectors $\phi(x_k, u_k)$ if $u_k = -Kx_k$?)
- Given θ_K, how do we compute the improved policy K'? Write out the resulting approximate policy iteration algorithm. (With LSTD-Q used for policy evaluation, this algorithm is called least-squares policy iteration, or LSPI.)

(c) Let $\mathcal{Q}^*$ be the optimal state-action value function. Show that

$$\mathcal{Q}^*(x,u) = \begin{bmatrix} x \\ u \end{bmatrix}^\mathsf{T} H^* \begin{bmatrix} x \\ u \end{bmatrix} + q^*,$$

where H^* and q^* can be expressed in terms of the solution P^* of the discounted Riccati equation

$$P = Q + \gamma A^\mathsf{T} P A - \gamma^2 A^\mathsf{T} P B (R + \gamma B^\mathsf{T} P B)^{-1} B^\mathsf{T} P A,$$

and find the optimal gain K^* (so that $u_k = -K^* x_k$ is optimal). (Hint: Use the optimal Bellman equation $\mathcal{Q}^*(x,u) = x^\mathsf{T} Q x + u^\mathsf{T} R u + \gamma\, \mathbb{E} \min_{u'} \mathcal{Q}^*(x', u')$ with $x' = Ax + Bu + w$.)

3. (20 points) **Policy Gradient**

Consider a nonlinear system $x_{k+1} = f(x_k, u_k, w_k)$ where x_k is the state, $u_k \in \mathbb{R}^p$ is the action, the initial state x_0 is sampled from a distribution $\mathcal{D}$, and the process noise w_k is sampled from a Gaussian distribution in an IID manner. The objective is to choose u_k to minimize the cost

$$\mathcal{C} = \mathbb{E} \sum_{k=0}^{\infty} \gamma^k c(x_k, u_k)$$

where $0 < \gamma < 1$ is the discount factor. Suppose f is unknown, and we want to learn a policy from sampled trajectories $\{(x_k, u_k, c(x_k, u_k))\}$.

(a) Consider a stochastic policy $\pi_\theta(u|x)$, i.e. a probability density over actions parameterized by θ, and let $\mathcal{C}(\theta)$ denote the cost when $u_k \sim \pi_\theta(\cdot|x_k)$. Let

$$\mathcal{Q}^\theta(x, u) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k c(x_k, u_k) \,\middle|\, x_0 = x,\, u_0 = u\right]$$

be the state-action value function of π_θ, where $u_k \sim \pi_\theta(\cdot|x_k)$ for $k \geq 1$. State the policy gradient theorem, i.e. give a formula for $\nabla_\theta \mathcal{C}(\theta)$ in terms of $\nabla_\theta \log \pi_\theta(u|x)$ and $\mathcal{Q}^\theta$.

(b) Suppose we use a three-layer neural network to parameterize the mean of a Gaussian policy:

$$u_k \sim \mathcal{N}\left(\mu_\theta(x_k),\, s^2 I_p\right), \qquad \mu_\theta(x) = W^2 \sigma(W^1 \sigma(W^0 x)),$$

where $\theta = (W^0, W^1, W^2)$, the activation function σ is differentiable and applied elementwise (e.g. tanh), and the standard deviation $s > 0$ is fixed.

- Write out an explicit formula for $\log \pi_\theta(u|x)$.
- Compute $\nabla_{W^2} \log \pi_\theta(u|x)$, and explain how the gradients with respect to W^1 and W^0 follow from the chain rule (backpropagation).
- Given N sampled trajectories of length T, write out a Monte Carlo estimate of $\nabla_\theta \mathcal{C}(\theta)$ based on (a), where $\mathcal{Q}^\theta(x_k, u_k)$ is replaced by the observed discounted cost-to-go $\sum_{t=k}^{T-1} \gamma^{t-k} c(x_t, u_t)$. Keep the factor γ^k in front of the k-th term. Briefly explain why truncating the trajectories at length T makes this estimate biased, and discuss qualitatively why the bias is small when T is large (you may assume that all the expectations involved are finite).

(c) Now suppose f is linear, i.e. $f(x_k, u_k, w_k) = Ax_k + Bu_k + w_k$ with $w_k \sim \mathcal{N}(0, W)$, and $c(x_k, u_k) = x_k^\mathsf{T} Q x_k + u_k^\mathsf{T} R u_k$. Suppose $\mathbb{E}[x_0 x_0^\mathsf{T}] = \Sigma_0 \succ 0$ and x_0 is independent of $\{w_k\}$.

Consider a deterministic linear policy $u_k = -Kx_k$, where K is stabilizing as in Problem 2(a). If $(A, B, Q, R, W, \Sigma_0)$ is known, we can calculate the gradient $\nabla \mathcal{C}(K)$ using the model information directly. Show that

$$\mathcal{C}(K) = \text{trace}\left(P_K\left(\Sigma_0 + \frac{\gamma}{1-\gamma}W\right)\right),$$

where P_K is defined in Problem 2(a). Then derive the gradient formula

$$\nabla \mathcal{C}(K) = 2\left((R + \gamma B^\mathsf{T} P_K B)K - \gamma B^\mathsf{T} P_K A\right)\Sigma_K,$$

where $\Sigma_K = \sum_{k=0}^{\infty} \gamma^k \, \mathbb{E}[x_k x_k^\mathsf{T}]$ satisfies $\Sigma_K = \Sigma_0 + \frac{\gamma}{1-\gamma}W + \gamma(A - BK)\Sigma_K(A - BK)^\mathsf{T}$.

4. (20 points) **Evolution Strategies**

The policy gradient method in Problem 3 explores by injecting noise into the actions. Evolution strategies (ES) instead learn a deterministic policy $u_k = \mu_\theta(x_k)$ by perturbing the policy parameter $\theta \in \mathbb{R}^d$ directly, and they only use evaluations of the cost $\mathcal{C}(\theta)$ (e.g. from rollouts). Throughout this problem, assume that $\mathcal{C} : \mathbb{R}^d \to \mathbb{R}$ is differentiable and $\nabla\mathcal{C}$ is L-Lipschitz, i.e. $\|\nabla\mathcal{C}(\theta) - \nabla\mathcal{C}(\theta')\| \le L\|\theta - \theta'\|$ for all θ, θ'. You may assume that all expectations below are finite, and that limits, derivatives, and expectations can be interchanged. ES uses the following quantity as an estimate of $\nabla\mathcal{C}(\theta)$:

$$g_\sigma(\theta) = \frac{1}{\sigma}\,\mathbb{E}_{\varepsilon\sim\mathcal{N}(0,I_d)}\left[\mathcal{C}(\theta+\sigma\varepsilon)\,\varepsilon\right],$$

where $\sigma > 0$ is the perturbation size.

(a) Show that subtracting the baseline $\mathcal{C}(\theta)$ does not change the mean, i.e.

$$\frac{1}{\sigma}\,\mathbb{E}_{\varepsilon\sim\mathcal{N}(0,I_d)}\left[(\mathcal{C}(\theta+\sigma\varepsilon) - \mathcal{C}(\theta))\,\varepsilon\right] = g_\sigma(\theta).$$

Then use the fact $\lim_{\sigma\to 0}\frac{\mathcal{C}(\theta+\sigma\varepsilon)-\mathcal{C}(\theta)}{\sigma} = \nabla\mathcal{C}(\theta)^\mathsf{T}\varepsilon$ to show that $\lim_{\sigma\to 0} g_\sigma(\theta) = \nabla\mathcal{C}(\theta)$.

(b) Define the Gaussian-smoothed cost $\mathcal{C}_\sigma(\theta) = \mathbb{E}\left[\mathcal{C}(\theta+\sigma\varepsilon)\right]$, where $\varepsilon \sim \mathcal{N}(0, I_d)$. Show that $\nabla\mathcal{C}_\sigma(\theta) = g_\sigma(\theta)$ for every $\sigma > 0$. Hence, ES performs exact gradient descent on the smoothed cost $\mathcal{C}_\sigma$ rather than on $\mathcal{C}$. (Hint: Since $\sigma\varepsilon$ has the density $p_\sigma(z) = (2\pi\sigma^2)^{-d/2}e^{-\|z\|^2/(2\sigma^2)}$, we can write $\mathcal{C}_\sigma(\theta) = \int_{\mathbb{R}^d}\mathcal{C}(y)\,p_\sigma(y-\theta)\,dy$. Differentiate this integral with respect to θ.)

5. (20 points) **Implementation Assignment**

In this problem, you are asked to implement LSPI from Problem 2(b) for an LQR example with the following parameters

$$A = \begin{bmatrix} 0.99 & 0.01 & 0 \\ 0.01 & 0.95 & 0.01 \\ 0.5 & 0.14 & 0.97 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0.1 \\ 0 & 0.2 \\ 1.1 & 0.7 \end{bmatrix}, \quad R = \begin{bmatrix} 1.3 & 0 \\ 0 & 0.7 \end{bmatrix},$$

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.5 \end{bmatrix}, \quad W = \begin{bmatrix} 0.3 & 0 & 0 \\ 0 & 0.15 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}.$$

Here W is the covariance matrix of the IID zero-mean Gaussian process noise w_k. We fix the discount factor as $\gamma = 0.98$. Notice that $K = 0$ is not stabilizing for this system in the sense of Problem 2(a) (check this). Use the initial policy

$$K_0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0.1 \end{bmatrix},$$

which satisfies $\rho(A - BK_0) < 1$, where $\rho(\cdot)$ denotes the spectral radius.

(a) Since the model is known here, compute P^* and K^* by solving the discounted Riccati equation in Problem 2(c) (e.g. by iterating the Riccati recursion until convergence). Report K^*. This serves as the ground truth for LSPI below.

(b) Generate a data set of N transitions (x_k, u_k, c_k, x_{k+1}) by simulating the system with $u_k = -K_0 x_k + e_k$, where $e_k \sim \mathcal{N}(0, I)$ is IID exploration noise. Use several rollouts of length 100, each starting from $x_0 \sim \mathcal{N}(0, I)$. Using this data set, implement LSPI from Problem 2(b) starting from K_0. At iteration $j \geq 1$, write the estimated $\mathcal{Q}$-function as

$$\mathcal{Q}_j(x, u) = \begin{bmatrix} x \\ u \end{bmatrix}^{\mathsf{T}} H_j \begin{bmatrix} x \\ u \end{bmatrix} + \text{constant}, \qquad H_j = \begin{bmatrix} H_j^{xx} & H_j^{xu} \\ H_j^{ux} & H_j^{uu} \end{bmatrix},$$

and let K_j be the resulting greedy gain. Run LSPI until $\|K_j - K_{j-1}\|_F$ falls below a tolerance or a maximum number of iterations is reached (report both choices). Plot the relative error $\|K_j - K^*\|_F / \|K^*\|_F$ versus j.

With finite data, LSPI can fail. At every iteration, your code should check the following:

- $H_j^{uu} \succ 0$. This guarantees that $\min_u \mathcal{Q}_j(x, u)$ has a unique minimizer, so the greedy gain is well defined. (If H_j^{uu} has a negative eigenvalue, then $\min_u \mathcal{Q}_j(x, u)$ is unbounded below.)

- K_j is stabilizing in the sense of Problem 2(a). Otherwise the cost of K_j is infinite, and K_j cannot be evaluated in the next iteration.

If a check fails, stop that run and report which check failed and at which iteration. A correctly diagnosed failure is an acceptable result.

(c) Repeat (b) for several values of N between 10^3 and 10^5. Since the results depend on the random data set, use at least 5 random seeds for each N and report the outcome of every run. Does LSPI work? Summarize your findings, e.g. how the accuracy depends on N, how many iterations LSPI needs, and any failures you observe. Submit your code.