Fall 2021 *Instructor: Vidya Muthukumar*

ECE 8803: Online Decision Making in Machine Learning

Homework 5

Released: Nov 11 *Due: Nov 24*

Objective. To gain hands-on experience with introductory topics in reinforcement learning and optimal control, including dynamic programming, sample complexity, and exploration.

Note: You only need to attempt 2 out of 3 problems in the HW. Your grade will constitute the 2 problems on which you did the best overall.

Problem 1 (Dynamic programming in linear quadratic control) 25 points At the beginning of our discussion of reinforcement learning, we introduced the dynamic programming algorithm. While our discussion was centered around discrete (finite-sized) state and action spaces, the DP principle can work much more generally for continuous state and action spaces, provided that the state-evolution and cost functions are highly structured. This allows the applicability of the DP principle to practical control systems. Throughout this problem, you will want to *minimize* cost.

In this problem, we will explore the DP principle, as applied to linear quadratic control. First, we specify the MDP as below:

- The state-space is given by $\mathcal{S} = \mathbb{R}^2$, and the action space is given by $\mathcal{A} = \mathbb{R}$. A physical interpretation is as follows: one dimension of the state space denotes position and the other denotes velocity. And the action (typically called the *control*) denotes an accelerative force that is applied.

- For a state $\boldsymbol{s} \in \mathcal{S}$ and an action $a \in \mathcal{A}$, the state evolution is *deterministic* and is given by

$$\boldsymbol{s}' = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \boldsymbol{s} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} a. \tag{1}$$

  This is commonly called a *linear dynamical system*, and commonly models the evolution of physical state spaces.

- For a state $\boldsymbol{s} \in \mathcal{S}$ and an action $a \in \mathcal{A}$, the immediate *cost* is equal to

$$c(\boldsymbol{s}, a) = \|\boldsymbol{s}\|_2^2 + a^2 \tag{2}$$

  This is commonly called a *quadratic cost* function, and penalizes both large state values (as these can be *unstable*) and large controls (as these are *expensive* to apply).

- We consider the finite-horizon model for cost. A policy is given by a mapping $\boldsymbol{\pi} := (\pi_1, \ldots, \pi_H)$ where each π_h maps $\mathcal{S}$ to $\mathcal{A}$ (i.e. $\pi_h : \mathbb{R}^2 \to \mathbb{R}$). We will only need consider *linear* controls of the form

$$\pi_h(s) = \langle \boldsymbol{a}_h, \boldsymbol{s} \rangle \tag{3}$$

  for some vector $\boldsymbol{a}_h$.

- For horizon length H, we evaluate the total cost as

$$J^{\boldsymbol{\pi}}(\boldsymbol{s}_1) := \sum_{h=1}^{H} c(\boldsymbol{s}_h, \pi_h(\boldsymbol{s}_h)). \tag{4}$$

Note that we do not need any expectations as the state-evolution in Equation (1) as well as the immediate costs are *deterministic*. We will assume that we begin at $\boldsymbol{s}_1 = \mathbf{0}$.

(a) As in class, start with the case $h = H$. Show that the optimal policy π_H^* for *any* state $\boldsymbol{s}_H$ is given by $\pi_H^*(\boldsymbol{s}_H) = 0$. Write this optimal policy in terms of a linear control vector $\boldsymbol{a}_H^*$ as in Equation (3); i.e. express $\pi_H^*(\boldsymbol{s}_H) = \langle \boldsymbol{a}_H^*, \boldsymbol{s}_H \rangle$. Also write down an expression (equality) for the *cost-to-go* $J_H^*(\boldsymbol{s}_H)$, as a function of the final state $\boldsymbol{s}_H$.

(b) Now, let us go to the case $h = H - 1$. For any state $\boldsymbol{s}_{H-1}$, write down an expression for the action-value function, defined as

$$Q_{H-1}^*(\boldsymbol{s}_{H-1}, a) = c(\boldsymbol{s}_{H-1}, a) + J_H^*(\boldsymbol{s}_H).$$

Your expression should only be a function of $\boldsymbol{s}_{H-1}$ and a. You can write the expression in vector notation or with separate elements, i.e. $\boldsymbol{s}_{H-1} = \begin{bmatrix} s_{H-1,1} \\ s_{H-1,2} \end{bmatrix}$.

(c) Now, use the special structure in the action-value function $Q_{H-1}^*(\boldsymbol{s}_{H-1}, a)$ to find the optimal policy $\pi_{H-1}^*(\boldsymbol{s}_{H-1})$ that *minimizes* $Q_{H-1}^*(\boldsymbol{s}_{H-1}, a)$. Then, argue that this policy is *linear*, i.e. that it can be represented as $\pi_{H-1}^*(\boldsymbol{s}_{H-1}) = \langle \boldsymbol{a}_{H-1}^*, \boldsymbol{s}_{H-1} \rangle$ for some control vector $\boldsymbol{a}_{H-1}^*$.

Hint: first use your expression to deduce that the action-value function is quadratic in a. Use this observation to compute the minimizing value.

(d) Use your expression above to provide an expression for the cost-to-go $J_{H-1}^*(\boldsymbol{s}_{H-1})$. Argue that the expression is quadratic in $\boldsymbol{s}_{H-1}$ by expressing it as

$$J_{H-1}^*(\boldsymbol{s}_{H-1}) = \|\boldsymbol{s}_{H-1}\|_2^2 + \|\boldsymbol{M_1 s}_{H-1}\|_2^2 + \|\boldsymbol{M_2 s}_{H-1}\|_2^2$$

for some 2×2 matrices $\boldsymbol{M_1}, \boldsymbol{M_2}$. Specify the matrices $\boldsymbol{M_1}, \boldsymbol{M_2}$ explicitly in your answer.

Note that there are many possible choices for $\boldsymbol{M_1}, \boldsymbol{M_2}$. Any choice that yields the correct equation is acceptable.

(e) Now consider $h = H - 2$, and write down an expression for the action-value function $Q_{H-2}(\boldsymbol{s}_{H-2}, a)$. Use your answer to part (d) to show that $Q_{H-2}(\boldsymbol{s}_{H_2}, a)$ is quadratic in a, and use this to show that the optimal policy $\pi_{H_2}^*(\boldsymbol{s}_{H-2})$ will be linear. You do not need to provide an explicit expression for the optimal policy.

(f) (BONUS – 5 points) Use the principle of backward induction to formally argue that the optimal policy will *always* be linear for any $h = 1, \ldots, H$.

Hint: consider any step h, and assume that the cost-to-go function $J_{h+1}^*(\boldsymbol{s}_{h+1})$ is quadratic in $\boldsymbol{s}_{h+1}$. Then, it suffices to show that a) the optimal policy $\pi_h^*(\boldsymbol{s}_h)$ will be linear in $\boldsymbol{s}_h$, and b) the ensuing cost-to-go function $J_h^*(\boldsymbol{s}_h)$ will be quadratic as well.

Problem 2 (Sample complexity of RL through Hoeffding bounds) 25 points In class, we have spent some time examining *model-based* approaches, which go through the following three-step procedure:

- Collect samples of state transitions $S'_1(s,a), \ldots, S'_m(s,a)$ for every state-action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$. For simplicity, let us assume that we already know the immediate rewards $r(s,a)$ for every state-action pair.
- Estimate an MDP from these samples, and
- Compute an optimal policy for this estimated MDP, and return it as the *estimated* optimal policy.

The total number of samples is equal to $n := m|\mathcal{S}||\mathcal{A}|$. We claimed in class that you require $\mathcal{O}\left(\frac{|\mathcal{S}||\mathcal{A}|\text{poly}(\frac{1}{1-\gamma})}{\epsilon^2}\right)$ samples in order to estimate an ϵ-optimal policy. In this problem, we will show that a *larger* number of samples: $\mathcal{O}\left(\frac{|\mathcal{S}|^3|\mathcal{A}|\text{poly}(\frac{1}{1-\gamma})}{\epsilon^2}\right)$, is sufficient to learn an ϵ-optimal policy using Hoeffding bounds. We will use the discounted MDP model for this problem.

(a) Consider a state-action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$. From the samples $S'_1(s,a), \ldots, S'_m(s,a)$ of the state transitions, and every value of $s' \in \mathcal{S}$, we can construct estimates of the transition probability entry $P(s'|s,a)$ as

$$\widehat{P}(s'|s,a) = \frac{1}{m}\sum_{i=1}^{m} \mathbb{I}[S'_i(s,a) = s']$$

Use Hoeffding's inequality to show that for a fixed tuple (s,a,s'), we have $|\widehat{P}(s'|s,a) - P(s'|s,a)| \leq \sqrt{\frac{\log(2/\delta)}{2m}}$ with probability at least $1-\delta$.

Hint: Note that you need to apply the upper and lower tail of Hoeffding's inequality for this sub-part.

(b) Take a union bound over all pairs (s,a,s') to show that

$$\max_{(s,a,s')} |\widehat{P}(s'|s,a) - P(s'|s,a)| \leq \sqrt{\frac{\log(2|\mathcal{S}|^2|\mathcal{A}|/\delta)}{2m}}$$

with probability at least $1-\delta$.

Hint: use an adjusted value of $\delta' := \frac{\delta}{|\mathcal{S}|^2|\mathcal{A}|}$ *and apply part (a) together with the union bound.*

(c) Now, we define the *on-policy* action-value functions $Q^\pi(s,a) := r(s,a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)}[V^\pi(s')]$, and the corresponding *estimated* on-policy action-value function[1] as $\widehat{Q}^\pi(s,a) := r(s,a) + \gamma \mathbb{E}_{s' \sim \widehat{P}(\cdot|s,a)}[V^\pi(s')]$. We will show that $|\widehat{Q}^\pi(s,a) - Q^\pi(s,a)|$ is small for all state-action pairs $(s,a) \in \mathcal{S} \times \mathcal{A}$, when the error in estimating the MDP is small. This suffices to show that

[1]We have made a idealized simplification in the estimate here: that the $V^\pi(s')$ in the next step is the same for both estimators. Don't worry about this simplification; it turns out not to really matter.

the estimated optimal policy is close to the true optimal policy. Use the definitions of the on-policy action-value functions to show that

$$|\widehat{Q}^{\pi}(s,a) - Q^{\pi}(s,a)| \leq \frac{1}{1-\gamma} \cdot \sum_{s' \in \mathcal{S}} |\widehat{P}(s'|s,a) - P(s'|s,a)|.$$

Hint: Use the fact that the immediate rewards are between $[0,1]$ *to show that the value function for any policy at any starting state, i.e.* $V^{\pi}(s)$*, is always bounded between* 0 *and* $\frac{1}{1-\gamma}$*.*

(d) Use the expressions in parts (b) and (c) to provide an upper bound on the estimation error $|\widehat{Q}^{\pi}(s,a) - Q^{\pi}(s,a)|$ as a function of $|\mathcal{S}|, |\mathcal{A}|, \gamma, \delta$ that holds with probability at least $1-\delta$. As a result, how many *total samples* do you need to ensure that $|\widehat{Q}^{\pi}(s,a) - Q^{\pi}(s,a)| \leq \epsilon$ with probability at least $1-\delta$? Express your answer as a function of $|\mathcal{S}|, |\mathcal{A}|, \gamma, \delta$ and the error tolerance ϵ.

Hint: look at the last steps in the appendix of Lecture Note 20 to gain some familiarity with the meaning of a sample complexity bound. (You do not need to look at the appendix for anything else.)

Problem 3 (Optimal control and RL for deep-sea exploration) 25 points In this problem, you will solve the deep-sea exploration problem that we alluded to in class, and you will see the benefits of exploration that prioritizes state visitation. The deep sea exploration MDP is specified as follows (we have used Python indexing that matches with the indexing provided in the starter iPython notebook, for convenience):

- N^2 states, where a state is represented as $s := (i, j)$ with $i, j \in \{0, 1, \ldots, N-1\}$. Here, i represents the *row* index ($i = 0$ is the top of Figure 1 and $i = N-1$ is the bottom of Figure 1). Similarly, j represents the *column* index ($j = 0$ is the left of Figure 1 and $j = N-1$ is the right of Figure 1). Consequently, $(i, j) = (0, 0)$ represents the top-left corner (marked with a sailboat), and $(i, j) = (N-1, N-1)$ marks the bottom-right corner (marked with a treasure chest).

- 2 actions for each state and deterministic transitions. As long as $i < N-1$, we have

$$P((i', j')|(i, j), 1) = \begin{cases} 1 \text{ if } i' = i+1, j' = \max\{j-1, 0\} \\ 0 \text{ otherwise} \end{cases} \quad \text{and}$$

$$P((i', j')|(i, j), 2) = \begin{cases} 1 \text{ if } i' = i+1, j' = \min\{j+1, N-1\} \\ 0 \text{ otherwise.} \end{cases}$$

  If $i = N-1$, we have

$$P((i', j')|(N-1, j), a) = \begin{cases} 1 \text{ if } i' = 0 \text{ and } j' = j \\ 0 \text{ otherwise.} \end{cases}$$

  Consequently, action 1 takes us left and downwards (or just downwards if we are already maximally left, i.e. at column 0). Action 2 takes us right and downwards (or just downwards if we are already maximally right, i.e. at column $N-1$). If we are at the bottom (maximally downwards), we go back to the top of the column we are in regardless of which action is taken.

- Deterministic rewards, given by

$$r((i, j), a) = \begin{cases} 0 \text{ if } a = 1 \\ -\frac{0.01}{N} \text{ if } a = 2 \end{cases} \quad \text{when } (i, j) \neq (N-1, N-1).$$

  For the bottom-right state $(i, j) = (N-1, N-1)$, we will consider two different types of reward: the *treasure reward* and the *bomb reward*. Under the treasure reward, we have $r((N-1, N-1), a) = 100$ for either value of $a = 1, 2$. Under the bomb reward, we have $r((N-1, N-1), a) = -100$ for either value of $a = 1, 2$. We will call the respective MDPs the *treasure MDP* and the *bomb MDP* respectively.

- Finite horizon of length $H = N = 5$.

You will find it useful to refer to Lecture Note 21 and the description of the E3 algorithm to answer this question. Also see the attached iPython notebook for starter code that specifies the MDP model.

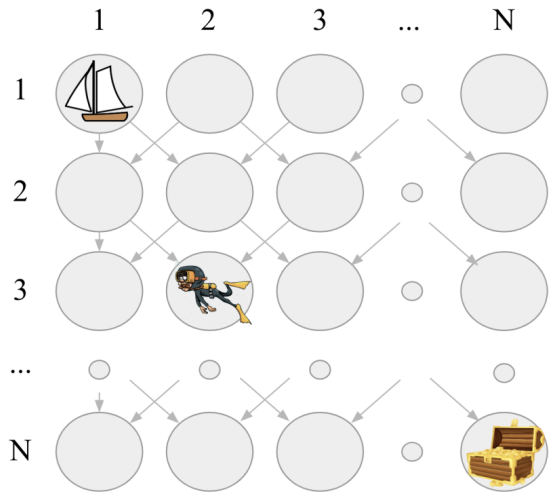

Figure 1: Schematic of the deep-sea environment. In this problem, you will solve the deep-sea exploration problem for depth $N = 5$.

(a) Implement the DP algorithm from class in order to compute the optimal policy for horizon length $H = N = 5$, beginning at the top-left state $s_1 = (0, 0)$, for both the treasure and bomb MDP. Write down the path that is induced by the optimal policy, and the corresponding total reward, in each case.

(b) Now, let us compare the "speed" of two types of exploration routines in RL. First, implement the E3 algorithm from class, again starting at the top-left state $(0, 0)$, with the following simplifications/specifications:

- When performing balanced wandering, break ties in favor of action 1 (i.e if $N_{(s_t,1)}(t) = N_{(s_t,2)}(t)$, pick action 1).
- The choice $n_{\min} = 1$ for all state-action pairs (which means that you designate a state $s \in \mathcal{S}_{\mathsf{known}}$ if you have seen all actions in that state at least once).
- When you reach a state $s_t \in \mathcal{S}_{\mathsf{known}}$, pick the action a_t that maximizes $\sum_{s' \notin \mathcal{S}_{\mathsf{known}}} \widehat{P}_t(s'|s_t, a)$ (where $\widehat{P}_t(s'|s_t, a)$ is your current estimate for the transitions at that state).
- Terminate the procedure once all of the *diagonal* states are known, i.e. $(i, i) \in \mathcal{S}_{\mathsf{known}}$ for all $i \in \{0, \ldots, N-1\}$.

Run your implementation for the "treasure" and "bomb" MDP. Report the number of steps that it takes to terminate, and the estimated MDP, in both cases. (Write your answer for the estimated MDP by specifying the estimated rewards and transitions for every state-action pair.)

(c) Use your answer for the estimated MDP to estimate the optimal policy, and report the difference, if any, between the estimated optimal policy and the true optimal policy.

Hint: the transitions and rewards are deterministic. Will there be any estimation error?

(d) (BONUS — 5 points) Consider a variant of the E3 algorithm with the same specifications in part (b) *except* that does not designate states as "known". In other words, this will *always* perform balanced wandering and will pick $a_t = \arg\min_{a \in \mathcal{A}} N_{(s_t,a)}(t)$ at round t (with the same tie-breaking rule as in part (b)). Terminate the procedure once all of the diagonal states have been visited at least once.

Report the number of steps that this variant takes to terminate for the treasure MDP only. How does it compare to E3?