Homework 1: Markov Processes and Dynamic Programming

Collaboration in the sense of discussion is allowed, however, the work you turn in should be your own - you should not split parts of the assignments with other students and you should certainly not copy other students' solutions or code. ***If you discuss your homework with someone, please indicate their name(s) in your submission****. See the collaboration and academic integrity statement here:* https://natanaso.github.io/ece276b*. Books may be consulted but not copied from.*

Submission

You should submit the following files on **Gradescope** by the deadline shown at the top right corner.

1. Upload your solutions to problems in pdf format: **FirstnameLastname_HW1.pdf**. You may use latex, scanned handwritten notes (write legibly!), or any other method to prepare a pdf file. Do not just write the final result. Present your work in detail, explaining your approach at every step.
2. Upload all **python** code you have written for each problem in zip format: **FirstnameLastname_HW1.zip**. The zip file should also include a README file with a clear description of the files used for each problem and how they can be run.

Problems

In square brackets are the points assigned to each problem.

1. [25 pts] Consider a Markov Chain with five states, $\mathcal{X} = \{1, 2, 3, 4, 5\}$ and the following transition matrix:

$$P = \begin{bmatrix} 1/2 & 1/4 & 0 & 0 & 1/4 \\ 1/4 & 1/2 & 1/4 & 0 & 0 \\ 0 & 1/4 & 1/2 & 1/4 & 0 \\ 0 & 0 & 1/4 & 1/2 & 1/4 \\ 1/4 & 0 & 0 & 1/4 & 1/2 \end{bmatrix} \tag{1}$$

   (a) Draw the state transition diagram for this chain (like the ones shown in the lecture). Is this Markov Chain irreducible? Is it periodic? Explain your answers.

   (b) What is the long-term behavior of this Markov chain? In other words, if this chain were initialized from an initial mass function $p_0 = \left[\frac{25}{150}, \frac{20}{150}, \frac{35}{150}, \frac{24}{150}, \frac{46}{150}\right]^T$, how would p_t evolve over time? Compare a simulated value of p_{100} to the theoretical value of $p_\infty := \lim_{t\to\infty} p_t$.

   (c) **The average age problem**. Consider a group of 5 people sitting at round table, in a way that each person can only talk with his/her right and left neighbor. The five people are aged 25, 20, 35, 24, and 46, respectively. Each person knows only their own age but can talk with his/her neighbors to obtain information from them. Based on your observations in parts a) and b) above, describe an approach for determining the average age of the five people, still under the assumption that the people can only talk with their neighbors.

2. [25 pts] Consider a system with the following motion model:

$$x_{t+1} = x_t u_t + u_t^2$$

   The system state x_t can only take on values $\{-1, 0, 1, 2\}$, while the control input u_t is constrained to be -1 or 1. Let the planning horizon be $T = 2$, stage cost be $\ell(x, u) := xu$, and terminal cost be $\mathfrak{q}(x) = x^2$.

   (a) Use dynamic programming to find an optimal policy.

   (b) Find the optimal cost, control sequences, and state trajectory for $x_0 = 2$.

3. [30 pts] You are controlling a linear system by selecting its modes of operation:

$$x_{t+1} = \begin{cases} Ax_t, & \text{if } u_t = 1, \\ Bx_t, & \text{if } u_t = 2 \end{cases}$$

where $x_t \in \mathbb{R}^n$ is the current state, and $A, B \in \mathbb{R}^{n\times n}$ are two given matrices. The stage and terminal costs of operating the system are the same and satisfy $\ell(x,u) \equiv \mathfrak{q}(x) := \frac{1}{2}x^T x$. This setup has applications in sensor scheduling and in embedded control systems with limited computation and communication resources.

   (a) Use dynamic programming to show that the optimal cost-to-go/value function $V_t^\star(x)$ of the problem is the minimum of 2^{T-t} positive definite quadratic functions. Describe the optimal policy $\pi_t^\star$ using these functions.

   (b) Consider the $T = 3$ horizon problem with matrices $A = \begin{bmatrix} 0.75 & -1 \\ 1 & 0.75 \end{bmatrix}$ and $B = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$. Plot the regions of the state space where it is optimal to select action 1 and 2 respectively at time $t = 0, \ldots, 3$, for example on a discretized $[-1, 1] \times [-1, 1]$ square around the origin. Also, plot the optimal value function $V_0^\star(x)$ on that space. Justify your plots through mathematical analysis.

4. [55 pts] In this problem, your task is to solve the deterministic shortest path problem on a given graph. The input to your program contains the number of nodes in the graph n, the start node s, the goal node τ, and a matrix $C \in \mathbb{R}^{n\times n}$ specifying the cost C_{ij} of transitioning from node i to node j. If a transition is not possible, $C_{ij} = \infty$. An instance of the problem is visualized in Fig. 1. Your solution should include

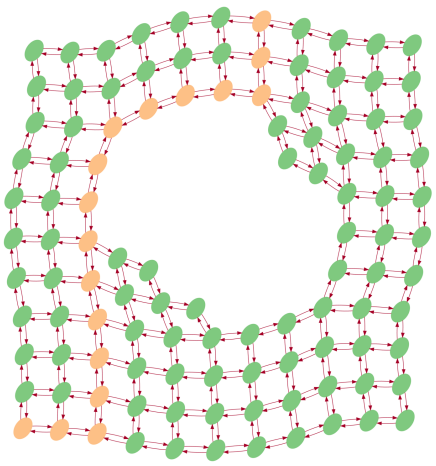

Figure 1: A deterministic shortest path problem with the minimum cost path from start to goal shown in orange. The edges specify the possible transitions among the nodes.

   the following components:

   (a) **Problem Formulation**: define the problem as a Markov Decision Process and clearly specify its elements, including the state and control spaces, the initial state, the motion model, the planning horizon, the stage and terminal costs, and any other elements necessary to make this a well defined problem.

   (b) **Technical Approach**: describe your approach for computing a minimum cost path in a deterministic shortest path problem. Write down the equations you would use explicitly. Implement your technical approach in python.

   (c) **Results**: Your results should contain a list of vertices along the minimum cost path (from the start to the goal vertex) as well as a list of the optimal cost-to-go values $(V^\star(x_0), V^\star(x_1), \ldots, V^\star(x_n))$ for each problem instance. The expected output for the first problem is shown in Fig. 1. The figure was

generated using the provided visualization code, which requires the `graphviz` package. You may generate such plots for the other problem instances. Note that your code will be tested on instances other than the provided ones. Make sure you follow the input/output conventions exactly.

5. [65 pts] Confident in your newfound dynamic programming skills, you challenge your friend to a 100 game match of rock-paper-scissors. Your friend is a good player and knows to randomize his strategy but he is still biased towards one of the three options. You are not certain if your friend prefers rock, paper, or scissors but you know that he plays his preferred move 50% of the time and each of the other two options, 25% of the time. Your goal is to maximize your score, i.e., beat your friend with as large of a lead as possible by the end of the 100th game.

POMDP

   (a) **Problem Formulation**: formulate this problem as a Markov Decision Process. Clearly define the state space $\mathcal{X}$, the control space $\mathcal{U}$, the motion model f or p_f, the initial state x_0, the planning horizon T, the stage and terminal costs ℓ, $\mathfrak{q}$, and any other elements necessary to make this a well defined problem. Note that your opponent's bias towards one of the options is not directly observable.

   (b) **Technical Approach**: describe how you would go about finding an optimal policy for maximizing the expected lead with which you beat your opponent. Write down the explicit equations you would use to solve this problem. Implement your theoretical idea in python.

   (c) **Results**: use your python implementation to compare three different strategies – *deterministic*, *stochastic*, and *optimal*. The deterministic strategy iterates rock, paper, scissors, rock, paper, scissors, rock,.... The stochastic strategy chooses among the three options uniformly at random for each game. The optimal strategy is the one you formulated and computed above. Provide a plot showing the mean and standard deviation over 50 100-game matches played by the three strategies with the number of games on the x axis, and the game score differential on the y axis. Assume here that your friend has a bias towards paper and generate 5000 plays from his strategy. Use the same data to compare the performance of the deterministic, stochastic, and optimal policies. Provide a discussion of your results.