Assignment 4 of MATP6960: programming

(Due at 11:59PM on Dec-15-2020)

Instruction: Each student needs to submit the source code file and a report by Latex. You are required to do both problems, each for 50% credit. **You will be evaluated based on whether you have all the results in the requirements below, and also the report**. Compress your source files (**DO NOT send the data.**) and PDF report file into a single .zip file, name it as "MATP6960_Assignment4_YourLastNameInitial", and send it to `optimization.rpi@gmail.com`

1 Parking Problem

In class, we talked about using dynamic programming to model a parking problem. Suppose there are N spaces in a parking area, and they are numbered $0, \ldots, N-1$. Also, assume there is a garage. The driver starts at space 0 and traverses the parking spaces sequentially. Each parking space k costs $c(k)$ and is free (i.e., not occupied) with probability $p(k)$. If the driver reaches the last parking space $N-1$ and does not part there, the driver must part in the garage that costs C.

Requirements

1. Introduce state and control variables and also a dynamic system to formulate the parking problem as a dynamic program. The formulation has been talked in class, so this question is to check how you understand the problem. You will just need to restate the modeling process.

2. For the following specific data:

$$C = 100, N = 200, p(k) = 0.05, c(k) = N - k, \forall\, k = 0, 1, \ldots, N-1,$$

   write a program either in MATLAB or Python or some other programming language to compute the optimal expected cost-to-go function value at each stage and at each state. In addition, give the optimal policy, namely, you need to give a decision rule when to park such that the expected cost is minimized.

2 Simulation-based evaluation of Q-factors

In class, we talked about simulation-based implementation of the rollout algorithm. Recall that the Q-factor is defined as

$$Q_k(x_k, u_k) = \mathbb{E}\Big[g_k(x_k, u_k, w_k) + J_{k+1,\pi}\big(f_k(x_k, u_k, w_k)\big)\Big]$$

where g_k is the cost function, and $J_{k+1,\pi}$ is the cost-to-go function by following the base policy π. With the Q-factors, then a rollout policy will choose the control by

$$\tilde{\mu}_k(x_k) \in \underset{u_k \in U_k(x_k)}{\arg\min}\, Q_k(x_k, u_k).$$

Practically, we may not know the model but can only observe the state and the value of g_k. Hence, we can only approximately evaluate Q_k by an empirical mean $\tilde{Q}_k$ through collecting samples.

1. Let the number of horizon be N. Suppose the state space $\mathcal{X}_k = \{1, 2, \ldots, 10\}$ at each stage $k = 0, \ldots, N-1$. The control space $U_k(x_k) = \{\text{L, R}\}$ if $x_k \neq 1, 10$, $U_k(x_k) = \{\text{R}\}$ if $x_k = 1$, and $U_k(x_k) = \{\text{L}\}$ if $x_k = 10$. Hence, you can imagine that on a straight rope, there are 10 knots, and 1 and 10 are the two end knots. One ant climbs on the rope. At each non-end knot, it can choose to go to left or right, but at the left end knot, it can only go to right, and at the right end knot, it can only go to left. The disturbance w_k depends on u_k and follows the following distribution:

$$\begin{aligned}
\text{Prob}(w_k = 0 | u_k = \text{L}) = 0.1, \quad \text{Prob}(w_k = -1 | u_k = \text{L}) = 0.4,\\
\text{Prob}(w_k = -2 | u_k = \text{L}) = 0.3, \quad \text{Prob}(w_k = -3 | u_k = \text{L}) = 0.2,\\
\text{Prob}(w_k = 0 | u_k = \text{R}) = 0.2, \quad \text{Prob}(w_k = 1 | u_k = \text{R}) = 0.3,\\
\text{Prob}(w_k = 2 | u_k = \text{R}) = 0.4, \quad \text{Prob}(w_k = 3 | u_k = \text{R}) = 0.1.
\end{aligned}$$

   The system follows the transition:

$$f_k(x_k, u_k, w_k) = \min(10, \max(1, x_k + w_k)).$$

   Define a base policy as $\pi(x_k) = \text{L}$ if $x_k \geq 6$ and $\pi(x_k) = \text{R}$ if $x_k \leq 5$.

   Write a function (in MATLAB or Python) to evaluate the Q-factors at all pairs of state and control for each stage. Treat the cost function g_k, the number of horizon N, and a budget of the number of samples as input of your function. For the provided data of g_k and N, compute the Q-factors by your function, and give a rollout policy (i.e., a sequence of maps from state to control).

2. [**up to 20% Bonus credit on top of the total credit**] We talked about adaptive sampling in class. Use adaptive sampling to evaluate the Q-factors. Note that the adaptive sampling will not accurately evaluate the Q-factors at all pairs of state and control but give more samples to the state-control pairs that potentially have small Q-factors. You can use the the current empirical mean for one pair of state and control as its exploitation index and the current sample number as the exploration index. Or you can design other quantities as the exploitation and exploration indices (and discuss the motivation of your design). For the provided data, test your adaptive sampling to see if the adaptive sampling can give the small Q-factors faster than the function you wrote in the first question, and also check if it gives the same rollout policy.