Contents

1 Before Lecture

-

2 Basic Ingredients of Stochastic Optimal Control

- Question: What where the 3 basic ingredients of a stochastic optimal control problem?
- Stochastic dynamical system:
- $x_{t+1} = f_t(x_t, u_t, w_t)$, $t = 0, 1, ..., T-1$
- $x_t \in \mathcal{X}_t$
- $u_t \in \mathcal{U}_t(x_t)$
- $w_t \sim \mathbb{P}_t^w(x_t, u, t)$
- $x_0 \sim \mathbb{P}_0$
- Design space for control laws/policies with an information pattern
- $\pi = \{\pi_0, \pi_1, ..., \pi_{T-1}\} \in \Pi$
- state feedback: $u_t = \pi_t(x_t)$, $\pi_t : \mathcal{X}_t \to \mathcal{U}_t$
- Temporally-additive cost function to be optimized
- $E\left[\sum_{t=0}^{T-1} g_t(x_t, u_t, w_t) + g_T(x_T)\right]$
- For a given initial state x_0 and a fixed control policy $\pi = \{\pi_0, ..., \pi_{T-1}\}$, the expected cost of π starting from x_0 is
- $J_\pi(x_0) \triangleq E\left[\sum_{t=0}^{T-1} g_t(x_t, \pi_t(x_t), w_t) + g_T(x_T)\right]$
- Goal: Find a state feedback policy that minimizes expected cost

- $J^*(x_0) = \min_{\pi \in \Pi} J_\pi(x_0)$
- $J^* : \mathcal{X} \to \mathbb{R}$ is a function that assigns to each initial state the optimal cost for starting from that state (called the optimal cost/value function)

3 Principle of Optimality

- Due to Richard Bellman, about 1950s
- A simple, easy to understand idea, that has significant implications, and is the main idea behind dynamic programming - which we will be using to solve stochastic optimal control problems
- Let $\pi^* = [\pi_0^*, \pi_1^*, ..., \pi_{T-1}^*]$ be an optimal policy.
- Assume that when using π^*, a given state x_i occurs at time step i with some positive probability. Then consider the subproblem where at time i, we are sitting at state x_i and we want to minimize the cost-to-go from time i to T
- $E\left[\sum_{t=i}^{T-1} g_t(x_t, \pi_t(x_t), w_t) + g_T(x_T)\right]$
- Then the truncated policy $[\pi_i^*, \pi_{i+1}^*, ..., \pi_{T-1}^*]$ is optimal for this tail subproblem.
- Activity: Think of a conceptual argument that proves this principle of optimality
- We are trying to prove a statement that says "if A (π^* is an optimal policy), then B (truncated policy $[\pi_i^*, \pi_{i+1}^*, ..., \pi_{T-1}^*]$ is optimal for this tail subproblem), $A \implies B$.
- Can potentially prove $A \implies B$ three ways:
- Direct Proof: Assume A is true and work to show that B must be true
- Contrapositive Proof: Assume B is false and work to show that A must be false
- Proof by contradiction: Assume A is true and B is false, then work to show an obvious contradiction of the form C is both true and false.
- Assume that B (truncated policy $[\pi_i^*, \pi_{i+1}^*, ..., \pi_{T-1}^*]$ is optimal for this tail subproblem) is false. Then there exists a better (lower cost) policy for the tail subproblem. Then you can simply substitute that better tail policy into the overall policy and get a better (lower cost) policy, which means A must be false.
- Draw: Show an example for travel through a graph.
- The principle of optimality suggests a decomposition to construct an optimal policy
- Piece together a policy working backwards in time.
- First solve the "tail subproblem" over the last stage
- Then solve the "tail subproblem" over the last two stages

- Keep going until you get back to time 0
- The stochastic optimal control problem has "optimal substructure with overlapping subproblems"
- The ability to decompose the problem can give a significant reduction in computational complexity.

4 The Dynamic Programming Algorithm

- For every initial state x_0, the optimal cost $J^*(x_0)$ is equal to $J_0(x_0)$ given by the last step of the following recursive algorithm, which goes backwards in time.
- Initialization: $J_T(x_T) = g_T(x_T)$
- Recursion: for $t = T-1, ..., 0$, $J_t(x_t) = min_{u_t \in \mathcal{U}_t(x_t)} E_{w_t} \left[g_t(x_t, u_t, w_t) + J_{t+1}(f_t(x_t, u_t, w_t))\right]$
- As a result, the optimal policy is given by $\pi^* = [\pi_0^*, \pi_1^*, ..., \pi_{T-1}^*]$ where
- $\pi_t^*(x_t) \in argmin_{u_t} E_{w_t} \left[g_t(x_t, u_t, w_t) + J_{t+1}(f_t(x_t, u_t, w_t))\right]$
- $\in$ because the minimizer might not be unique
- Key idea: at each time step, we need to optimize for all possible states $x_t \in \mathcal{X}_t$, this point-wise optimization results in functions for $J_t(x_t)$ and $\pi_t^*(x_t)$.
- Optimal input at time t balances the expect current stage cost with the expected future cost for the corresponding next state under an optimal control policy
- By storing cost-to-go functions, we avoid repeating computations.
- Activity: Lets demonstrate the idea of dynamic programming in a deterministic case of traversing a checkers board. Each space has a cost associated with traveling through that space. We can chose to start anywhere on the first row and we want to get to the last row with the least total cost. We can only move straight, diagonally left, and diagonally right, and we can't fall off the board. How do we determine the path of lowest cost?

5 Summary

- Dynamic Programming is an extremely powerful solution algorithm that theoretically works for any standard stochastic optimization problem
- However, the solution is abstract and cannot be computationally implemented in general
- We will focus on 2 spectial cases where DP leads to a computationally tractable algorithm in practice
  - Finite state, input, and disturbance spaces - DP results in basic vector/array operations
  - Linear Quadratic problems (linear dynamics, quadratic cost, continuous vector spaces) - DP results in basic linear algebra operations

6 Action Items (5 minutes)

- Work on HW #1