Dynamic Programming and Optimal Control

Study Note

Author: Jinyi Liu

Institute: Haslam Business School, UTK

Date: October 28, 2022

Bio: A first-year Ph.D. student in Business Analytics.

Victory won't come to us unless we go to it.

Contents

Chapter 1 The Dynamic Programming Algorithm

1.1 Introduction

1.1.1 General Structure of Finite Horizon Optimal Control Problems

Our finite horizon model has two principal features: (1) a *discrete-time dynamic system,* and (2) a *cost function that is additive over time.* The system has the form

$$x_{k+1} = f_k(x_k, u_k, w_k), \quad k = 0, 1, \dots, N-1,$$

where

$$\begin{array}{r|l} x_k & \text{state variable} \\ u_k & \text{control variable} \\ w_k & \text{random parameter,} \end{array}$$

and f_k is a function the describes the system.

The cost function is additive. The total cost is

$$g_N(x_N) + \sum_{i=0}^{N-1} g_k(x_k, u_k, w_k).$$

Since w_k is random, we formulate the problem as an optimization of the *expected cost*

$$E\left\{ g_N(x_N) + \sum_{i=0}^{N-1} g_k(x_k, u_k, w_k) \right\}.$$

1.2 The Basic Problem

Basic Problem

We are given a discrete-time dynamic system

$$x_{k+1} = f_k(x_k, u_k, w_k), \quad k = 0, 1, \dots, N-1,$$

where the state $x_k \in S_k$, the control $u_k \in C_k$ and the random "disturbance" w_k is an element of a space D_k.

The control u_k is constrained to be $u_k \in U_k(x_k) \subset C_k$ for all $x_k \in S_k$ and k.

w_k is characterized by a probability distribution $P_k(\cdot | x_k, u_k)$ that may explicitly on x_k and u_k but not on values of prior disturbances $w_{k-1}, \dots, w_0$.

We consider the class of polices

$$\pi = \{\mu_0, \dots, \mu_{N-1}\}$$

, where μ_k maps x_k into controls $u_k = \mu_k(x_k)$ and is such that $\mu_k(x_k) \in U_k(x_k)$ for all $x_k \in S_k$. Such polices will be called *admissible.*

Given x_0 and admissible π, we have

$$x_{k+1} = f_k(x_k, \mu_k(x_k), w_k), \quad k = 0, 1, \ldots, N-1 \tag{1.1}$$

Thus, for given function g_k, we have the expected cost of π starting at x_0:

$$J_\pi(x_0) = E\left\{ g_N(x_N) + \sum_{i=0}^{N-1} g_k(x_k, \mu_k(x_k), w_k) \right\}$$

where the expectation is taken over x_k and w_k. An optimal policy π^* is one such that

$$J_{\pi^*}(x_0) = \min_{\pi \in \Pi} J_\pi(x_0).$$

The Role and Value of Information

Encoding Risk in the Cost Function

1.3 The Dynamic Programming Algorithm

The DP algorithm rests on the *principle of optimality*.

The DP Algorithm

Proposition 1.3.1

For every initial state x_0, the optimal cost $J^*(x_0)$ of the basic problem is equal to $J_0(x_0)$, given by the last step of the following algorithm, which proceeds backward in time from period $N-1$ to period 0 :

$$J_N(x_N) = g_N(x_N),$$

$$J_k(x_k) = \min_{u_k \in U_k(x_k)} \mathop{E}_{w_k} \left\{ g_k(x_k, u_k, w_k) + J_{k+1}(f_k(x_k, u_k, w_k)) \right\}$$

$$k = 0, 1, \ldots, N-1,$$

where the expectation is taken with respect to the probability distribution of w_k, which depends on x_k and u_k. Furthermore, if $u_k^* = \mu_k^*(x_k)$ minimizes the right side of Eq. (1.6) for each x_k and k, the policy $\pi^* = \{\mu_0^*, \ldots, \mu_{N-1}^*\}$ is optimal.

1.4 State Augmentation and Other Reformulations

The general guideline in *state augmentation* is to *include in the enlarged state at time k all the information that is known to the controller at time k and can be used with advantage in selecting u_k.*

Time Delays

1.5 Some Mathematical Issues

Well-defined random variables.

1.6 Dynamic Programming and Minimax Control

Consider a triplet (Π, W, J), where π is the set of policies under consideration, W is the set in which the uncertain quantities are known to belong, and $J : \Pi \times W \to [-\infty, +\infty]$ is a given cost function. The objective is to

$$\min \max_{w \in W} J(\pi, w)$$

over all $\pi \in \Pi$.

Lemma 1.6.1

Let $f : W \to X$ be a function, and M be the set of all functions $\mu : X \to U$, where W, X, and U are some sets. Then for any functions $G_0 : W \to (-\infty, \infty]$ and $G_1 : X \times U \to (-\infty, \infty]$ such that

$$\min_{u \in U} G_1(f(w), u) > -\infty, \quad \textit{for all } w \in W,$$

we have

$$\min_{\mu \in M} \max_{w \in W} \left[G_0(w) + G_1(f(w), \mu(f(w)))\right] = \max_{w \in W} \left[G_0(w) + \min_{u \in U} G_1(f(w), u)\right].$$

Chapter 2 Deterministic Systems and the Shortest Path Problem

In this chapter we focus on deterministic problems, i.e., w_k can take only one value. In contrast with stochastic problems, *using feedback results in no advantage in terms of cost reduction.*

2.1 Finite-State Systems and Shortest Paths

The DP algorithm takes the form

$$J_N(i) = a_{it}^N, \quad i \in S_N, \tag{2.1}$$

$$J_k(i) = \min_{j \in S_{k+1}} \left[a_{ij}^k + J_{k+1}(j) \right], \quad i \in S_k, \quad k = 0, 1, \ldots, N-1. \tag{2.2}$$

A Forward DP Algorithm for Shortest Path Problems

An optimal path from s to t is also an optimal path from t to s in a "reverse" shortest path problem. It is given by

$$\tilde{J}_N(j) = a_{sj}^0, \quad j \in S_1, \tag{2.3}$$

$$\tilde{J}_k(j) = \min_{i \in S_{N-k}} \left[a_{ij}^{N-k} +_{k+1} (i) \right], \quad j \in S_{N-k+1}, \quad k = 1, 2, \ldots, N-1. \tag{2.4}$$

The optimal cost is

$$\tilde{J}_0(t) = \min_{i \in S_N} \left[a_{ij}^N + \tilde{J}_1(i) \right].$$

The above equations yield the same result

$$J_0(s) = \tilde{J}_0(t).$$

Note that there is no analog of forward DP algorithm for stochastic problems.

Converting a Shortest Path Problem to a Deterministic Finite-Stage Problem

2.2 Some Shortest Path Applications

2.2.1 Critical Path Analysis

2.2.2 Hidden Markov Models and the Vaterbi Algorithm

We are given the probability $r(z; i, j)$ of an observation taking value z when the state transition is from i to j. We assume independent observations; i.e., an observation depends only on its corresponding transition and not on other transitions. Time independent. π initial state's probability.

Given the observation sequence $Z_N = \{z_1, z_2, \ldots, z_N\}$, we adopt $\hat{X}_N = \{\hat{x}_0, \hat{x}_1, \ldots, \hat{x}_N\}$ that maximizes over all $X_N = \{x_1, x_2, \ldots, x_N\}$ the conditional probability $p(X_N|Z_N)$. This is called the *maximum a posteriori probability* approach.

Using independence, we have

$$p(X_N, Z_N) = \pi_{x_0} \prod_{k=1}^{N} p_{x_{k-1}x_k} r(z_k; x_{k-1}, x_k) \tag{2.5}$$

Trellis diagram and Viterbi algorithm.

The problem of maximizing $p(X_N, Z_N)$ is equivalent to the problem

$$\text{minimize } -\ln(\pi_{x_0}) - \sum_{k=1}^{N} \ln\left(p_{x_{k-1}x_k} r\left(z_k; x_{k-1}, x_k\right)\right)$$

over all possible sequences $\{x_0, x_1, \ldots, x_N\}$.

2.3 Shortest Path Algorithms

To be continued.

Chapter 3 Problesm with Perfect State Information

In this chapter we consider a number of applications of discrete-tiem stochastic optimal control with perfect state infomation.

3.1 Linear Systems and Quadratic Cost

In this section we consider the special case of a linear Systems

$$x_{k+1} = A_k x_k + B_k u_k + w_k, \quad k = 0, 1, \ldots, N-1,$$

and the quadratic cost

$$\underset{w_k, k=0,1,\ldots,N-1}{E} \left\{ x_N' Q_N x_N + \sum_{k=0}^{N-1} (x_k' Q_k x_k + u_k' R_k u_k) \right\}.$$

We assume that Q_k are postive semidefinite symmetric and R_k are positive definite symmetric. w_k has 0 mean and finite second moment.

Applying the DP algorithm, we have

$$\begin{aligned} J_N(x_N) &= x_N' Q_N x_N, \\ J_k(x_k) &= \min_{u_k} E\{x_k' Q_k x_k + u_k' R_k u_k + J_{k+1}(A_k x_k + B_k u_k + w_k)\}. \end{aligned} \tag{3.1}$$

We have the optimal control law for every k:

$$\mu_k^*(x_k) = L_k x_k, \tag{3.2}$$

where

$$L_k = -(B_k' K_{k+1} B_k + R_k)^{-1} B_k' K_{k+1} A_k,$$

and where the symmetric positive semidefinite matrices K_k are given by

$$K_N = Q_N, \tag{3.3}$$

$$K_k = A_k'(K_{k+1} - K_{k+1} B_k (B_k' K_{k+1} B_k + R_k)^{-1} B_k' K_{k+1}) A_k + Q_k \tag{3.4}$$

The optimal cost is then given by

$$J_0(x_0) = x_0' K_0 x_0 + \sum_{k=0}^{N-1} E\{w_k' K_{k+1} w_k\}.$$

The Riccati Equation and Its Asymptotic Behavior

Eq. (3.4) is called the *discrete-time Riccati equation*. If the matrices are constant, then as $k \to \infty$, K satisfies the *algebraic Riccati equation*

$$K = A'(K - KB(B'KB + R)^{-1} B'K)A + Q. \tag{3.5}$$

This property indicates that for a large N, one can approximate the control law Eq. (3.2) by $\{\mu^*, \ldots, \mu^*\}$, where

$$\mu^*(x) = Lx, \tag{3.6}$$

$$L = -(B'KB + R)^{-1}B'KA,$$

and K solves Eq. (3.5). This control law is *stationary*.

Definition 3.1.1

A pair (A, B), where A is an $n \times n$ matrix and B is an $n \times m$ matrix, is said to be controllable if the $n \times nm$ matrix

$$\left[B, AB, A^2B, \ldots, A^{n-1}B\right]$$

has full rank (i.e., has linearly independent rows). A pair (A, C), where A is an $n \times n$ matrix and C an $m \times n$ matrix, is said to be observable if the pair (A', C') is controllable, where A' and C' denote the transposes of A and C, respectively. ♣