Chapter 10: Dynamic Programming - Lecture Notes

Geometry of Optimal Decisions & Financial Applications

Lecture Notes

2026-01-20

Why Dynamic Programming Exists At All

Dynamic programming (DP) is not just an algorithm; it is a **principle of optimality**. It answers the fundamental question of sequential decision-making:

> "If I'm making decisions over time, how do I avoid being clever now and stupid later?"

The Principle of Optimality: An optimal strategy has the property that whatever state you end up in, the remaining decisions must themselves be optimal.

DP replaces brute-force enumeration with **structure**, describing problems in terms of **states**, **controls**, and **value functions**.

The Shortest Path Problem: DP in Its Purest Form

The shortest path problem is DP with all distractions removed.

- **State:** Your current node.
- **Control:** Which arc you take next.
- **Cost-to-go (Value Function):** Shortest remaining distance to the destination.

Bellman's Equation (Deterministic/Finite):

$$V_i = \min_j \{c_{ij} + V_j\}$$

Where: - c_{ij} is the immediate cost. - V_j is the optimal cost from neighbor j to the end.

> **Key Logic:** The past does not matter once you know the present node. We solve **backward**, not forward.

Deterministic Sequential Decision Processes

Adding time and continuous states:

- **System Dynamics:** $x_{t+1} = h_t(x_t, u_t)$
- **Objective (Additive):** $\sum_{t=0}^{T-1} f_t(x_t, u_t) + F_T(x_T)$

Bellman Equation:

$$V_t(x) = \min_u \{f_t(x, u) + V_{t+1}(h_t(x, u))\}$$

This shift moves us from searching for a *sequence* of controls to searching for a **function**: the value function.

The Curse of Dimensionality

DP's greatest obstacle: as the state space dimension increases, computational complexity multiplies exponentially.

> **Reframing:** The real problem is not optimizing controls; it is **approximating the value function**.

Stochastic Dynamic Programming

When uncertainty enters:

$$x_{t+1} = h(x_t, u_t, \varepsilon_{t+1})$$

Stochastic Bellman Equation:

$$V_t(x) = \min_u \{f(x, u) + \mathbb{E}_t[V_{t+1}(x_{t+1})]\}$$

- **Expectations** replace future certainty.
- Decisions must be **non-anticipative** (no clairvoyance).

American Options: DP Meets Monte Carlo

American options are the canonical DP problem in finance: - You decide *when* to exercise. - Exercise is irreversible. - The decision depends on future opportunities.

Bellman Recursion for Options:

$$V_j(S) = \max\{\text{intrinsic value},\ \mathbb{E}[e^{-r\Delta t} V_{j+1}(S_{j+1}) \mid S_j = S]\}$$

Least-Squares Monte Carlo (LSMC)

The Longstaff–Schwartz method (LSMC) is approximate DP in action:

1. **Simulate Paths:** Forward in time.
2. **Work Backward:** From expiration.
3. **Regression:** Approximate the **Continuation Value** by regressing discounted future payoffs against current states (using basis functions).
4. **Compare:** Exercise value vs. Continuation value.
5. **Update:** Cash flows and repeat.

> **Regression as Function Approximation:** We turn an infinite-dimensional problem into a parameter estimation problem to approximate the conditional expectation.

Big Picture: Why This Matters

The unifying idea across deterministic optimization, stochastic control, and financial derivatives is: **Optimal behavior over time = Value function + Backward reasoning.**

> **Core Shift:** Never try to optimize decisions directly when you can optimize a value function instead. Actions are temporary; values are the foundation of sequential rationality.