

Data-Driven Decision-Making In Enterprise Applications

Dynamic Programming

Rainer Schlosser

Hasso Plattner Institute (EPIC)

May 20, 2019

Outline

- Questions: Homework

Updated Code Examples

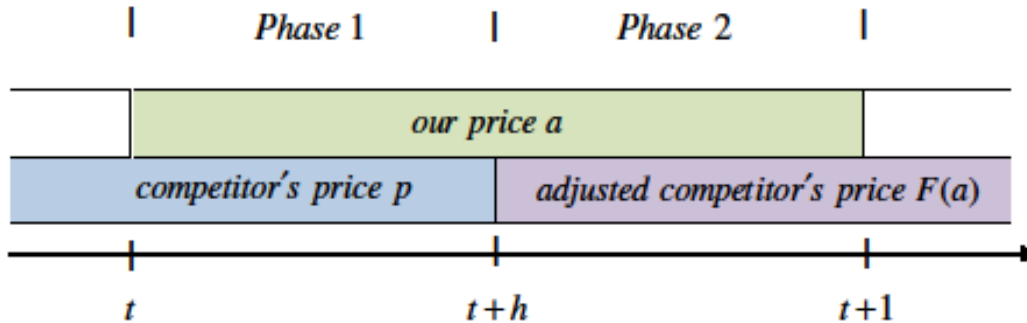
- Today: Learn about Dynamic Programming

Dynamic Pricing in a Duopoly

Motivation: Dynamic Pricing in a Duopoly

- Assume 2 sellers (duopoly). Assume only one feature: price
- Define different price reaction strategies $a(p)$, i.e.,
if the competitor's current price is p , we adjust our price to $a(p)$
Admissible prices are $a(p) \in \{1, 2, \dots, 100\}$
- Let the competitor's response strategy be given by: $p(a) := \max(a - 1, 1)$
- We adjust our prices a at times $t = 1, 2, 3, \dots$
The competitor adjusts his prices p at times $t = 0.5, 1.5, 2.5, \dots$

Sequence of Events (Duopoly Example)



- In every interval $(t, t + 0.5)$, $t = 0, 0.5, 1.0, \dots$, a sale occurs with probability $1 - \min(a_t, p_t) / 100$. With probability $\min(a_t, p_t) / 100$ no sale takes place
- If a sale takes place, our firm sells the item with probability

$$P(1, a, p) = 1_{\{a < p\}} + 0.5 \cdot 1_{\{a = p\}}, \text{ i.e., } P(0, a, p) = 1_{\{a > p\}} + 0.5 \cdot 1_{\{a = p\}} = 1 - P(0, a, p)$$

Duopoly Example

- What is a good counter strategy to be played against $p(a) := \max(a - 1, 1)$?
- Simulate different strategies $p(a)$, e.g., until time $T=1000$.
Start with $a_0 = p_0 = 20$ at time $t = 0$
- Which strategy $a(p)$ performs best, i.e., maximizes expected revenues?
- How to find the optimal strategy?
- Any ideas?

Goals for Today

- We want to optimally solve the duopoly example
- We have a dynamic optimization problem
- What are dynamic optimization problems?
- How to apply dynamic programming techniques?

What are Dynamic Optimization Problems?

- How to control a dynamic system over time?
- Instead of a single decision we have a sequence of decisions
- The system evolves over time according to a certain dynamic
- The decisions are supposed to be chosen such that a certain objective/quantity/criteria is optimized
- Find the right balance between short and long-term effects

Examples Please!

Examples

- Inventory Replenishment
- Reservoir Dam
- Drinking at a Party
- Exam Preparation
- Brand Advertising
- Used Cars
- Eating Cake

Task: Describe & Classify

- Goal/Objective
- State of the System
- Actions
- Dynamic of the System
- Revenues/Costs
- Finite/Infinite Horizon
- Stochastic Components



Classification

Example	Objective	State	Action	Dynamic	Payments
Inventory Mgmt.	min costs	#items	#order	entry-sales	order/holding
Reservoir Dam	provide power	#water	#production	rain-outflow	none
Drinking at Party*	max fun	%	#beer	impact-rehab	fun/money
Exam Preparation*	max mark/effort	#learned	#learn	learn-forget	effort, mark
Advertising	max profits	image	#advertise	effect-forget	campaigns
Used Cars	min costs	age	replace(y/n)	aging/faults	buy/repair costs
Eating Cake*	max utility	%cake	#eat	outflow	utility

* Finite horizon

General Problem Description

- What do you want to minimize or maximize (Objective)
- Define the state of your system (State)
- Define the set of possible actions (state dependent) (Actions)
- Quantify event probabilities (state+action dependent) (Dynamics) (!!)
- Define payments (state+action+event dependent) (Payments)
- What happens at the end of the time horizon? (Final Payment)

Dynamic Pricing Scenario (Duopoly Example)

- We want to sell items in a duopoly setting with finite horizon
- We can observe the competitor's prices and adjust our prices (for free)
- We can anticipate the competitor's price reaction
- We know sales probabilities for various situations
- We want to maximize total expected profits

Problem Description (Duopoly Example)

- Framework: $t = 0, 1, 2, \dots, T$ Discrete time periods
- State: $s = p$ Competitor's price
- Actions: $a \in A = \{1, \dots, 100\}$ Offer prices (for one period of time)
- Dynamic: $P(i, a, s)$ Probability to sell i items at price a
- Payments: $R(i, a, s) = i \cdot a$ Realized profit
- New State: $p \xrightarrow{\Gamma(i, a, s)} F(a)$ State transition / price reaction F
- Initial State: $s_0 = p_0 = 20$ Competitor's prices in $t=0$

Problem Formulation

- Find a *dynamic pricing strategy* that

maximizes total expected (discounted) profits:

- $$\max E \left[\sum_{t=0}^T \underbrace{\delta^t}_{\substack{\text{discount} \\ \text{factor}}} \cdot \left(\sum_{i_t \geq 0} \underbrace{P(i_t, a_t, S_t)}_{\substack{\text{probability to sell } i_t \text{ items} \\ \text{at price } a_t \text{ in situation } S_t}} \cdot \underbrace{i_t}_{\substack{\text{sales}}} \cdot \underbrace{a_t}_{\substack{\text{price}}} \right) \middle| \underbrace{S_0 = s_0}_{\substack{\text{initial state}}} \right], \quad 0 < \delta \leq 1$$

- What are admissible policies? Answer: Feedback Strategies
- How to solve such problems? Answer: Dynamic Programming

Solution Approach (Dynamic Programming)

- What is the **best expected value** of having the chance to sell . . .

“items from time t on starting in market situation s ”?

- Answer: That’s easy $V_t(s)$! ??????
- We have renamed the problem. Awesome. But - that’s a solution approach!
- We don’t know the “**Value Function V** ”, but V has to satisfy the relation

Value (state today) = Best expected (profit today + Value (state tomorrow))

Solution Approach (Dynamic Programming)

- *Value (state today) = Best expected (profit today + Value (state tomorrow))*
- Idea: Consider the transition dynamics within one period

What can happen during one time interval?

state today	#sales	profit	state tomorrow	probability	with (a, p) vs. $(a, F(a))$
	0	0	$F(a)$	$P(0, a, s)$	
$s = p$	1	a	$F(a)$	$P(1, a, s)$	
	2	$2a$	$F(a)$	$P(2, a, s)$	

- What does that mean for the **value** of state $s = p$, i.e., $V_t(s) = V_t(p)$?

Bellman Equation

state today	#sales	profit	state tomorrow	probability	with (a, p) vs. $(a, F(a))$
	0	0	$F(a)$	$P(0, a, s)$	
$s = (p)$	1	a	$F(a)$	$P(1, a, s)$	
	2	$2a$	$F(a)$	$P(2, a, s)$	

$$V_t(p) = \max_{a \geq 0} \left\{ \underbrace{P(0, a, p)}_{\text{probability not to sell}} \cdot \left(\underbrace{0 \cdot a}_{\text{today's profit}} + \underbrace{\delta \cdot V_{t+1}(F(a))}_{\text{best disc. exp. future profits}} \right) \right. \\
 \left. + \underbrace{P(1, a, p)}_{\text{probability to sell}} \cdot \left(\underbrace{1 \cdot a}_{\text{today's profit}} + \underbrace{\delta \cdot V_{t+1}(F(a))}_{\text{best disc. exp. future profits}} \right) + \dots \right\}$$

Optimal Solution

- We finally obtain the Bellman Equation:

$$V_t(p) = \max_{a \geq 0} \left\{ \sum_{i \geq 0} \underbrace{P(i, a, p)}_{\text{probability}} \cdot \left(\underbrace{i \cdot a}_{\text{today's profit}} + \underbrace{\delta \cdot V_{t+1}(F(a))}_{\text{best disc. exp. future profits of new state}} \right) \right\}$$

- Ok, but why is that interesting?
- Answer: Because $a_t^*(p) = \arg \max_{a \in A} \{ \dots \}$ is the *optimal policy*
- Ok! Now, we just need to compute the Value Function!

How to solve the Bellman Equation?

- Using the terminal condition $V_T(p) := 0$ for time horizon T (e.g., 1000)

We can compute the value function *recursively* $\forall t, p$:

$$V_t(p) = \max_{a \geq 0} \left\{ \sum_{i \geq 0} \underbrace{P(i, a, p)}_{\text{probability}} \cdot \left(\underbrace{i \cdot a}_{\text{today's profit}} + \underbrace{\delta \cdot V_{t+1}(F(a))}_{\text{best disc. exp. future profits of new state}} \right) \right\}$$

- The optimal strategy $a_t^*(p)$, $t = 1, \dots, T$, $p = 1, \dots, 100$, is determined by the arg max of the value function

How to solve the Bellman Equation?

- Using the terminal condition $V_T(p) := 0$ for time horizon T (e.g., 1000)

We can compute the value function *recursively* $\forall t, p$:

$$V_t(p) = \max_{a \geq 0} \left\{ \sum_{i \geq 0} \underbrace{P(i, a, p)}_{\text{probability}} \cdot \left(\underbrace{i \cdot a}_{\text{today's profit}} + \underbrace{\delta \cdot V_{t+1}(F(a))}_{\text{best disc. exp. future profits of new state}} \right) \right\}$$

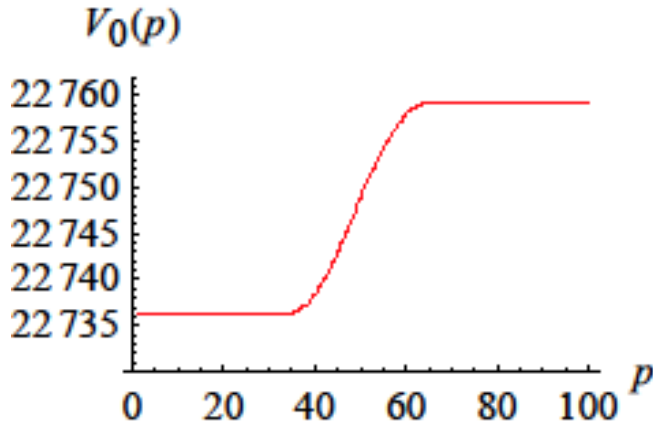
- The optimal strategy $a_t^*(p)$, $t = 1, \dots, T$, $p = 1, \dots, 100$, is determined by the arg max of the value function

- In AMPL:

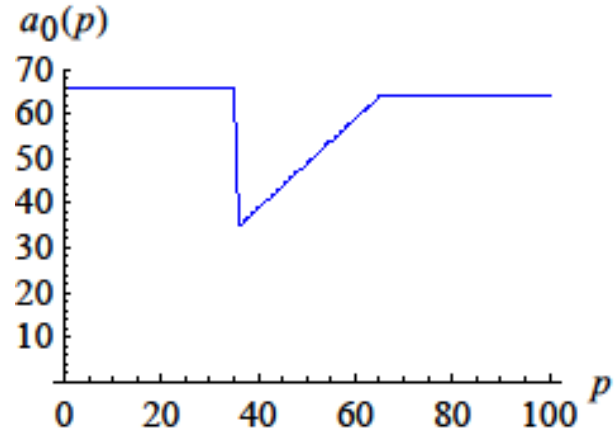
```
param V{t in 0..T,p in A}:= if t<T then max {a in A}
sum{i in I} P[i,a,p] * ( i*a + delta * V[t+1,F[a]] );
```

Optimal Pricing Strategy of the Duopoly Example

Best expected profit



Optimal pricing strategy



Infinite Horizon Problem

- $$\bullet \max E \left[\sum_{t=0}^{\infty} \underbrace{\delta^t}_{\text{discount factor}} \cdot \left(\sum_{i_t \geq 0} \underbrace{P(i_t, a_t, S_t)}_{\substack{\text{probability to sell } i_t \\ \text{at price } a_t \text{ in situation } S_t}} \cdot \underbrace{i_t}_{\text{sales}} \cdot \underbrace{a_t}_{\text{price}} \right) \underbrace{S_0 = s_0}_{\text{initial state}} \right], \underline{0 < \delta < 1}$$
- Will the value function $V_t^*(p)$ be time-dependent?
- Will the optimal price reaction $a_t^*(p)$ strategy be time-dependent?
- How does the Bellman equation look like?

Solution of the Infinite Horizon Problem

- $$V^*(p) = \max_{a \geq 0} \left\{ \sum_{i \geq 0} \underbrace{P(i, a, p)}_{\text{probability}} \cdot \left(\underbrace{i \cdot a}_{\text{today's profit}} + \underbrace{\delta \cdot V^*(F(a))}_{\text{disc. exp. future profits of new state}} \right) \right\}$$
- Approximate solution:** Finite horizon approach (value iteration)
- For “large” T the values $V_0(p)$ converge to the exact values $V^*(p)$
- The optimal policy $a^*(p)$, $p = 1, \dots, 100$, is determined by the arg max of the last iteration step, i.e., $a_0(p)$

Exact Solution of the Infinite Horizon Problem

- $$V^*(p) = \max_{a \geq 0} \left\{ \sum_{i \geq 0} \underbrace{P(i, a, p)}_{\text{probability}} \cdot \left(\underbrace{i \cdot a}_{\text{today's profit}} + \underbrace{\delta \cdot V^*(F(a))}_{\text{disc. exp. future profits of new state}} \right) \right\}, p \in A$$
- We have to solve a system of nonlinear equations
- Solvers can be applied, e.g., MINOS (see NEOS Solver)
- In AMPL:

```

subject to NB {p in A}: V[p] = max {a in A}
sum{i in I} P[i,a,p] * ( i*a + delta * V[F[a]] ); solve;

```

Questions?

- State
- Action
- Events
- Dynamics & State Transitions
- Recursive Solution Principle
- General concept applicable to other problems

Overview

2	April 25	Linear Programming I
3	April 29	Linear Programming II
4	May 2	Linear/Logistic Regression + Homework (3 weeks time)
5	May 16	Exercise Implementations
6	May 20	Dynamic Programming
7	May 23	Demand Learning & Price Optimization & Market Simulations
8	May 27	Project Assignments + Homework 2
9	June 3	Robust Optimization
10	June 13	Workshop / Group Meetings
11	June 20	Presentations (First Results)
12/13	June 24/27	Workshop / Group Meetings
14/15	July 1/4	Workshop / Group Meetings
16	July 11	Presentations (Final Results), Feedback, Documentation (Aug 31)