Data-Driven Demand Learning and Dynamic Pricing Strategies in Competitive Markets

Dynamic Programming

Rainer Schlosser, Martin Boissier, Matthias Uflacker

Hasso Plattner Institute (EPIC)

May 23, 2017

Outline

- Questions/Support: Data-Driven Pricing (Exercise)
- Goals of Today's Meeting: Solution of the Duopoly Game
- Learn about Dynamic Programming

Mandatory Exercise – Combine all Components

(1) **Create** random market situations with multiple sellers

Choose randomized prices for our firm (exploration phase)

(2) Choose a specific Buying Behavior, e.g., Approach II (with 0.6, 0.3, 0.1)

Simulate our firm's sales for all market situations

(3) **Estimate** sales probabilities, e.g., Logit model or Poisson via least squares

Use different combinations of explanatory variables

Mandatory Exercise – Combine all Components

(4) **Measure** the goodness of fit of your models, i.e.,

Compare original and estimated sales probabilities

(5) **Create** new random market situations with multiple sellers

Evaluate your estimated sales probabilities for potential offer prices

Compute prices that maximize expected short-term profits

(6) **Simulate** sales for all new market situations and your optimized prices

Compare realized profit for rule-based strategies & the optimized prices

Goals for Today

- We want to 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: $t = 0,1,2,\ldots,T$ — Discrete time periods
- State: $s = p$ — Competitor's price
- Actions: $a \in A = \{1,\ldots,100\}$ — 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:

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

- 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	$rf.\ (a,p)\ vs.\ (a,F(a))$
	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
	0	0	$F(a)$	$P(0,a,s)$
$s=(p)$	1	a	$F(a)$	$P(1,a,s)$
	2	$2a$	$F(a)$	$P(2,a,s)$

$rf.\ (a,p)\ vs.\ (a,F(a))$

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

$$\left.+\underbrace{P(1,a,p)}_{probability\ to\ sell}\cdot\left(\underbrace{1\cdot a}_{today's\ profit}+\underbrace{\delta\cdot V_{t+1}\left(F(a)\right)}_{best\ disc.\ exp.future\ profits}\right)+\ldots\right\}$$

Optimal Solution

- We finally obtain the Bellman Equation:

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

- Ok, but why is that interesting?

- Answer: Because $a_t^*(p) = \arg\max_{a \in A} \{...\}$ is the *optimal policy*

- Ahhh! 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* $\forall t, p$:

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

- The optimal strategy $a_t^*(p)$, $t = 1,...,T$, $p = 1,...,100$,

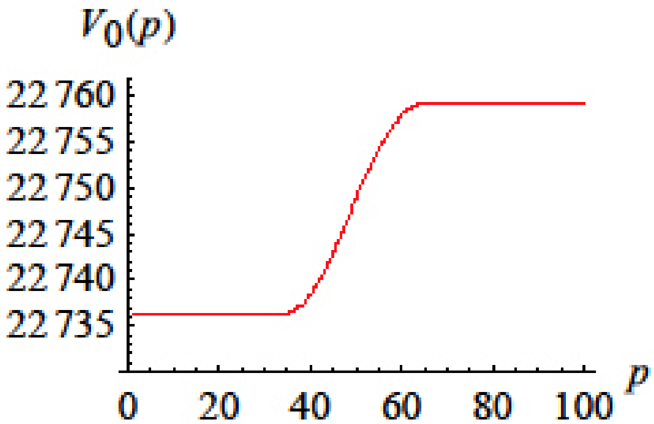
  is determined by the arg max of the value function

- In AMPL:

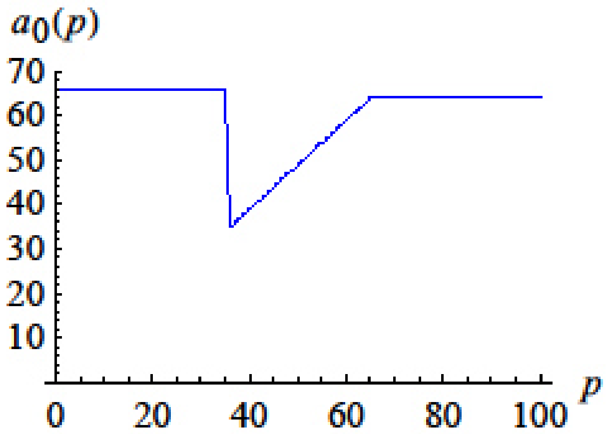
```
param V{t in 0..T,p in A}:= if t<T then max {a in A}
sum{i in I} P[i,a,p] * ( i*a + delta * V[t+1,F[a]] );
```

Optimal Pricing Strategy of the Duopoly Example

Best expected profit



Optimal pricing strategy



Infinite Horizon Problem

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

- Will the value function be time-dependent?

- Will the optimal price reaction strategy be time-dependent?

- How does the Bellman equation look like?

Solution of the Infinite Horizon Problem

- $$V^*(p) = \max_{a \geq 0} \left\{ \sum_{i \geq 0} \underbrace{P(i,a,p)}_{probability} \cdot \left(\underbrace{i \cdot a}_{today's\ profit} + \underbrace{\delta \cdot V^*\left(F(a)\right)}_{disc.\ exp.future\ profits\ of\ new\ state} \right) \right\}$$

- **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 $a^*(p)$, $n = 1, \ldots, N$, is determined by

  the arg max of the last iteration step, i.e., $a_0(p)$

Exact Solution of the Infinite Horizon Problem

- $$V^*(p) = \max_{a \geq 0} \left\{ \sum_{i \geq 0} \underbrace{P(i,a,p)}_{probability} \cdot \left(\underbrace{i \cdot a}_{today's\ profit} + \underbrace{\delta \cdot V^*\big(F(a)\big)}_{disc.\ exp. future\ profits\ of\ new\ state} \right) \right\},\ p \in A$$

- 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;
```

Recommended Exercise – Apply Dynamic Programming

- Solve the duopoly example with finite/infinite horizon
- Play with parameters (horizon, discount, sales probabilities, reaction times)
- Play with the competitor's strategy
- Apply demand learning (do not assume to know the true demand)
- Try to learn the competitor's strategy (do not assume to know it)

Overview

2	April 25	Customer Behavior
3	May 2	Demand Estimation
4	May 9	Pricing Strategies
5	May 16	no Meeting
6	May 23	Dynamic Programming (Optimal Solution of the Duopoly Game)
7	**May 30**	**Introduction Price Wars Platform &Dynamic Pricing Challenge**
8	June 6	Workshop / Group Meetings
9	June 13	Presentations (First Results)
10	June 20	Workshop / Group Meetings
11	June 27	no Meeting
12	July 4	Workshop / Group Meetings
13	July 11	Workshop / Group Meetings
14	**July 18**	**Presentations (Final Results), Feedback, Documentation (Aug/Sep)**

Price Wars Platform: Student Job

We are looking for a student assistant that works with us on our pricing platform.
We would like to extend our platform to cover additional scenarios in the future (e.g., restricted merchant storage, perishable goods, etc.) and get help during the seminar.

We are looking for a web-affine student that feels comfortable with the following technology stack:

- microservice-based architecture using RESTful HTTP communication
- a mix of services written in Ruby, Scala, Python, and Bash
- the main component: a merchant written in Python that we would like to improve

You're interested? Talk to us or write us a mail. :)