Dynamic Pricing on Online Marketplaces: Strategies and Applications

Dynamic Optimization Techniques

Dr. Rainer Schlosser

Hasso Plattner Institute (EPIC)

April 26, 2016

Outline

- Follow-Up: Demand Estimation
- Implementation of Regressions

- Goals of Today’s Meeting
- Dynamic Programming Problems

- Understand the Bellman Equation
- Solve the Bellman Equation

- Illustrations

Follow-Up: Demand Estimation

- Questions?
- Slides online
- Selection of Explanatory Variables
- First Implementations?
- Verification of your Model
- Example

Simulate & Verify your Model

- Idea: Use random numbers to simulate features x and observed sales y

- Example: $a_i = Uniform(1,20)$, $i = 1,...,10K$ our prices for $10K$ observations

  $p_{i,c} = Uniform(1,20)$, $c = 1,...,5$ prices of 5 competitors

- Sales: $y_i = round\left(Uniform(0,1-r(a_i,\vec{p}_i)/11)\right)$, **assume** a dynamic

  or $y_i = if\ Uniform(0,1) < e^{b'x_i} / (1+e^{b'x_i})\ then\ 1\ else\ 0$ (b known)

- Check if your regression finds the right coefficients / sales probabilities

- When your model can learn different dynamics – it can learn the true one!

Dynamic Optimization

Goals for Today

- We want to compute optimized prices.
- We want to be able to solve dynamic optimization problems.
- We want to understand the Bellman Equation.
- We want to learn how to solve the Bellman Equation.

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 decisions are supposed to be chosen such that
  a certain objective/quantity/criteria is optimized.
- The system evolves according to a certain dynamic.
- Dealing with the control of processes usually means
  to find the right balance between short and long term effects.

Examples Please!

- Used Cars
- Inventory Replenishment
- Reservoir Dam
- Drinking at a Party
- Exam Preparation
- Advertising
- Jerome Boateng
- Eating Cake
- Preparing Lectures

Describe & Classify

- Goal/Objective
- State of the System
- Actions
- Dynamic of the System
- Revenues/Costs
- Finite/Infinite Horizon
- What's stochastic?

Classification

Example	**Objective**	**State**	**Action**	**Dynamic**	**Payments**
Used Cars					
Inventory Mgmt.					
Reservoir Dam					
Drinking at Party					
Exam Preparation					
Advertising					
Jerome Boateng					
Eating Cake					
Lecture					

Problem Description

• Quantify what 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

- We want to sell durable goods (used books) on Amazon.
- We can have more than one item of the same book.
- We can observe competitors' prices and adjust our prices (for free).
- We can estimate sales probabilities for various situations.
- We have to take inventory holding costs into account (l/item/unit of time).
- We want to maximize total expected discounted profits, i.e.,
  the present value of expected sales revenues minus holding costs.

Problem Description (Basic Pricing Model)

- Framework: $t = 0, 1, 2, \ldots, T = \infty$ Discrete time periods, one type of book
- State: $s = (n, \vec{p})$ Number items left + competitors' prices
- Actions: $a \in A = 0.01, \ldots, 200$ Offer prices (for one period of time)
- Dynamic: $P(i, a, s)$ Probability to sell *i* items at price *a*
- Payments: $D(i, a, s) = i \cdot a - n \cdot l$ Sales revenue – Holding costs
- New State: $(n, \vec{p}) \to (n - i, F(a, \vec{p}))$ State Transition, Price Reactions *F*
- Initial State: $s_0 = (N, \vec{p}_0)$ *N* items + competitors' prices in *t*=0

Problem Formulation

- Find a Dynamic Pricing Policy that

  maximizes total expected discounted profits, i.e.,

- $$\max E\left[\sum_{t=0}^{\infty} \underbrace{\delta^t}_{\substack{discount\\factor}} \cdot \left(\sum_{i_t} \underbrace{P(i_t, a_t, S_t)}_{\substack{sales\ probability\\in\ market\ situation\ S}} \cdot \underbrace{a_t}_{price} - \underbrace{l \cdot X_t}_{\substack{holding\ costs\\X\ items\ left}} \right) \middle| \underbrace{S_0 = s_0}_{initial\ state} \right],$$

  where X_t is the random number of items left at time *t*.

- What are admissible policies? How to solve the problem? Ideas?

Solution Approach (Value Function)

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

“ n items in a market situation $\vec{p}$ ”?

- Answer: That’s easy $V(n, \vec{p})$! ?????

We have renamed the problem. Yeah! But wait - that’s a solution approach, because then the “Value Function” V has an implicit representation. ???

- We don’t know that function, but shouldn’t V satisfy the following:

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

Solution Approach (Bellman Equation)

- Value (state today) = best expected (profit today + Value (state tomorrow))

- Idea: Just consider the transition dynamics within one period.

  What can happen during one interval?

state today	#sales	profit	state tomorrow	probability
	0	$0-n \cdot l$	$(n-0, F(a, \vec{p}))$	$P(0, a, s)$
$s=(n, \vec{p})$	1	$a-n \cdot l$	$(n-1, F(a, \vec{p}))$	$P(1, a, s)$
	2	$2a-n \cdot l$	$(n-2, F(a, \vec{p}))$	$P(2, a, s)$

- What does that mean for the Value of state $s=(n, \vec{p})$?

Bellman Equation

state today	#sales	profit	state tomorrow	probability
	0	$0-n\cdot l$	$(n,\vec{p})$	$P(0,a,\vec{p})$
$s=(n,\vec{p})$	1	$a-n\cdot l$	$(n-1,\vec{p})$	$P(1,a,\vec{p})$
	2	$2a-n\cdot l$	$(n-2,\vec{p})$	$P(2,a,\vec{p})$

$$V(n,\vec{p})=\max_{a\geq 0}\left\{\underbrace{P(0,a,\vec{p})}_{probability\ not\ to\ sell}\cdot\left(\underbrace{0\cdot a}_{today's\ profit}-\underbrace{n\cdot l}_{\substack{holding\\costs}}+\underbrace{\delta\cdot V\left(n-0,F(a,\vec{p})\right)}_{disc.\ exp.future\ profits\ (n\ items)}\right)\right.$$

$$\left.+\underbrace{P(1,a,\vec{p})}_{probability\ to\ sell}\cdot\left(\underbrace{1\cdot a}_{today's\ profit}-\underbrace{n\cdot l}_{\substack{holding\\costs}}+\underbrace{\delta\cdot V\left(n-1,F(a,\vec{p})\right)}_{disc.\ exp.future\ profits\ (n-1\ items)}\right)+\ldots\right\}$$

Bellman Equation

- We finally obtain:

$$V(n,\vec{p}) = \max_{a\geq 0}\left\{\sum_i \underbrace{P(i,a,\vec{p})}_{probability}\cdot\left(\underbrace{\min(n,i)\cdot a}_{today's\ profit} - \underbrace{n\cdot l}_{\substack{holding\\ costs}} + \underbrace{\delta\cdot V\big(\max(0,n-i),F(a,\vec{p})\big)}_{disc.\ exp.future\ profits\ (n-1)^+\ items}\right)\right\}$$

- Ok, but why is that interesting?

- Answer: Because $a^*(n,\vec{p}) = \arg\max_{a\geq 0}\{...\}$ is the optimal policy.

- Ahhh. Now I want to know how to compute the Value Function!

How to solve the Bellman Equation?

- The Value function can be computed by . . .

		+	–
(i)	solving a system of nonlinear equations	exact	NLsolver needed
(ii)	solving a bigger system of linear equations	exact	many constraints
(iii)	recursive approximations	robust	not exact

- Note, in our dynamic pricing model under competition the state space can become very large.
- We will focus on (iii) recursive approximations.

Dealing with the Curse of Dimensionality

- Unfortunately, we have one major problem:

  The state space is too large (10 competitors -> #states 20.000^{10+1}).

- Next problem: The reaction functions of competitors are usually not known.

- That's why competitive pricing is challenging. But maybe there is hope:

  (i) In case of sticky/stable prices, cf. $F(a, \vec{p}) = \vec{p}$,
  the value function can be computed for single situations $\vec{p}$.

  (ii) Scenarios with just few active competitors can be handled.

Recursive Approximation of the Value Function

- Use the condition $V_t(0,\vec{p})=0$ (run-out) and consider the recursion:

$$V_t(n,\vec{p})=\max_{a\geq 0}\left\{\sum_i \underbrace{P(i,a,\vec{p})}_{probability}\cdot\left(\underbrace{\min(n,i)\cdot a}_{today's\ profit}-\underbrace{n\cdot l}_{\substack{holding\\costs}}+\underbrace{\delta\cdot V_{t+1}\left(\max(0,n-i),\vec{p}\right)}_{disc.\ exp.future\ profits\ (n-i)^+\ items}\right)\right\}$$

- Initialization: Choose T (e.g., 1000) and let $V_T(n,\vec{p}):=0$.

- For T "large" $V_0(n,\vec{p})$ converges to the exact value $V^*(n,\vec{p})$.

- The optimal policy $a^*(n,\vec{p})$, $n=1,...,N$,

  is determined by the arg max of the last iteration step.

Result: Feedback Pricing Strategies

- Prices can be easily computed for any other set of prices $\vec{p}$.

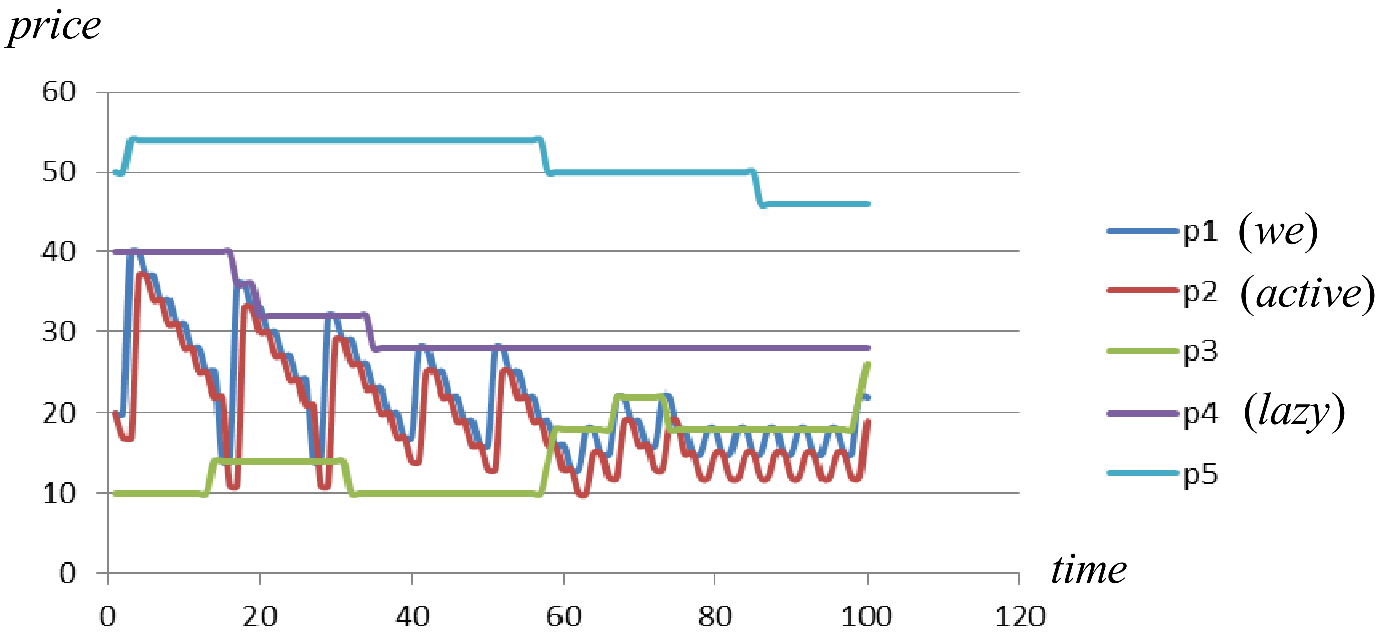
	n=1	n=2	n=3	n=4	n=5	n=7	n=10	n=20
price a^*	9.47	9.47	8.27	8.27	5.95	5.95	5.95	5.17
rank $r(a^*, \vec{p})$	5	5	4	4	2	2	2	1
$V^*(n, \vec{p})$	5.13	9.20	12.60	15.55	18.17	22.26	27.37	39.89

- Example of optimal feedback prices and expected profits,

$$\vec{p} = (5.18,\ 5.96,\ 6.31,\ 8.28,\ 9.48,\ 9.88,\ 10.33,\ 10.98,\ 11.67,\ 13.52)$$

- What are economic insights? Are they plausible?

Application: Dynamic Pricing under Competition



- Although we assume sticky prices, our policy reacts on changing market conditions!

Preview: Extensions & Dynamic Pricing Projects

- Time Dependence (e.g., Day of the Week)
- Finite Horizon / Perishable Products (e.g., Airline Tickets)
- Optimal Price Responses to Active Competitors (Price Anticipation)
- Multiple Products (Substitution Effects)
- Multiple Features (Ratings, Condition)
- Exit / Entry of Competitors, Strategic Customers, Advertising, Risk, . . .

Overview

2	April 19	Demand Estimation
3	April 26	Optimization Techniques
4	**May 3**	**Extensions / Projects**
5	May 10	Assign Projects to Teams
6	May 17	no Meeting
7	May 24	Workshop / Group Meetings
8	May 31	Workshop / Group Meetings
9	June 7	Presentations (First Results)
10	June 14	Workshop / Group Meetings
11	June 21	Workshop / Group Meetings
12	June 28	Workshop / Group Meetings
13	July 5	no Meeting
14	**July 12**	**Presentations (Final Results), Discussions, Feedback**