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

Introduction Price Wars Platform

Rainer Schlosser, Martin Boissier, Matthias Uflacker

Hasso Plattner Institute (EPIC)

May 30, 2017

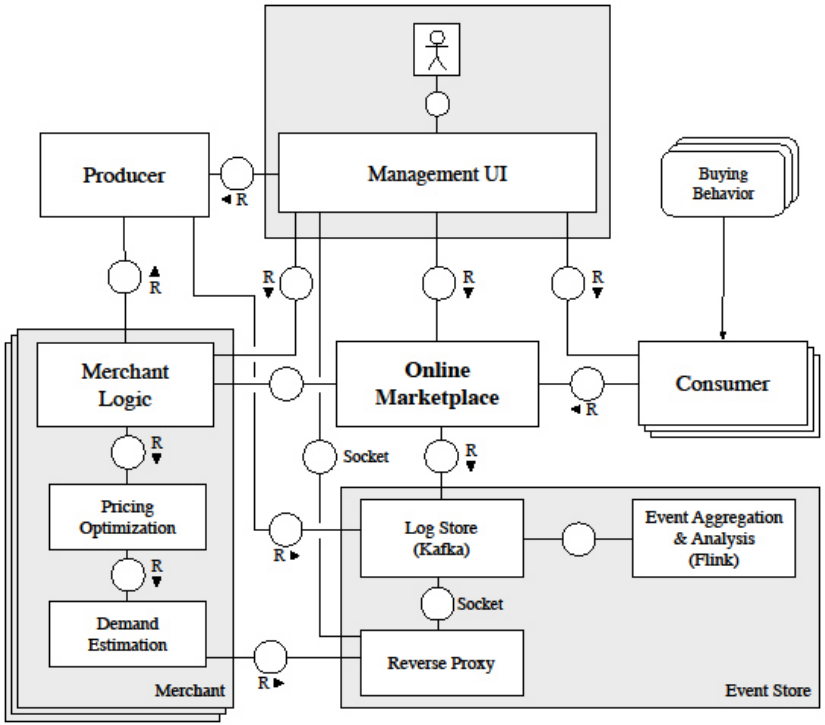
Outline

- Questions/Support: Data-Driven Pricing (Exercise)
- Questions/Support: Dynamic Programming (Exercise)
- Introduction Price Wars Platform
- Dynamic Pricing Challenge
- First Steps

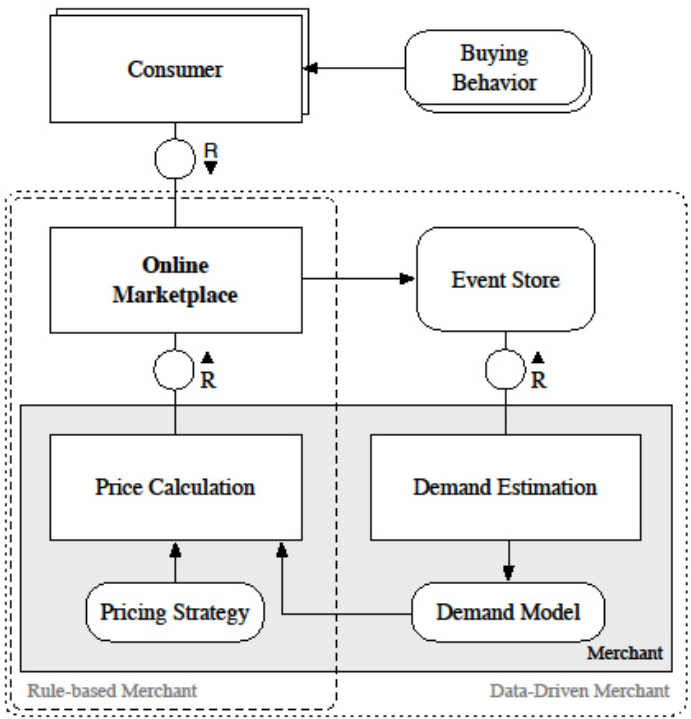
Goals for Today

- We want to play “dynamic pricing competition”
- How does our pricing simulation platform work?
  https://github.com/hpi-epic/masterproject-pricewars
- What will be expected from you?
- What kind of scenarios/setups will be considered?

How does our pricing simulation platform work?



Pricing Strategies and Demand Learning



Outlook: Dynamic Pricing Challenges

- Setup: Duopoly + Oligopoly, Features: Price + Quality
- Customer behavior: known + unknown
- Competitors' strategies: rule-based + data-driven
- Competitors' strategies: known + unknown
- Student teams play against each other

Start Simple: First Steps

- Customer behavior is known (only price impact)
- Understand event data (csv file) and join relevant data
- Create suitable explanatory variables
- Apply logistic regression (via Python)
- Check/verify your demand learning results
- Compute sales probabilities & expected profits

Market Situations Raw Data

amount,merchant_id,offer_id,price,prime,product_id,quality,shipping_time_prime,shipping_time_standard,timestamp,triggering_merchant_id,uid
...
1,4MRZ5eoIHSGs1GvLio2XKGksUFZIlTOcdy24mbnwq5c=,14304,18.0,True,3,2,1,5,2017-05-29T09:34:39.895Z,DaywOe3qbtT3C8wBBSV+zBOH55DVz40L6PH1/1p9xCM=,32
2,lsP4d66epeRdGEIB51N3sRN3Gy0R0b8qK+4rxc/EYqM=,14509,17.8,True,3,1,1,5,2017-05-29T09:34:39.895Z,DaywOe3qbtT3C8wBBSV+zBOH55DVz40L6PH1/1p9xCM=,31
1,hPjEe9kUnPadEcs0jO1HLUL5maZPb6umcWgcbCxHzdo=,14511,17.6,True,3,3,1,5,2017-05-29T09:34:39.895Z,DaywOe3qbtT3C8wBBSV+zBOH55DVz40L6PH1/1p9xCM=,33
1,DaywOe3qbtT3C8wBBSV+zBOH55DVz40L6PH1/1p9xCM=,14267,13.4,True,3,4,1,5,2017-05-29T09:34:39.895Z,DaywOe3qbtT3C8wBBSV+zBOH55DVz40L6PH1/1p9xCM=,14
3,4MRZ5eoIHSGs1GvLio2XKGksUFZIlTOcdy24mbnwq5c=,14345,13.5,True,3,3,1,5,2017-05-29T09:34:40.669Z,DaywOe3qbtT3C8wBBSV+zBOH55DVz40L6PH1/1p9xCM=,13
1,4MRZ5eoIHSGs1GvLio2XKGksUFZIlTOcdy24mbnwq5c=,14303,9.0,True,1,4,1,5,2017-05-29T09:34:40.669Z,DaywOe3qbtT3C8wBBSV+zBOH55DVz40L6PH1/1p9xCM=,14
2,4MRZ5eoIHSGs1GvLio2XKGksUFZIlTOcdy24mbnwq5c=,14329,18.0,True,1,2,1,5,2017-05-29T09:34:40.669Z,DaywOe3qbtT3C8wBBSV+zBOH55DVz40L6PH1/1p9xCM=,12

Note: First entry “amount” represents an inventory level and can be ignored.

Market Situations Raw Data (Selection)

merchant_id	price	product_id	quality	timestamp
A	18.0	3	2	2017-05-29T09:34:39.895Z
B	17.8	3	1	2017-05-29T09:34:39.895Z
C	17.6	3	3	2017-05-29T09:34:39.895Z
D	13.4	3	4	2017-05-29T09:34:39.895Z
A	13.5	3	3	2017-05-29T09:34:40.669Z
A	9.0	1	4	2017-05-29T09:34:40.669Z
A	18.0	1	2	2017-05-29T09:34:40.669Z

Recall notation: A seller's offer price a within a market situation $\vec{s}$ for a specific product

$(a, \vec{s}) = (\texttt{our price (A), timestamp, our quality, comp. price+quality})$

Observable Sales Events Raw Data

amount,consumer_id,http_code,left_in_stock,merchant_id,offer_id,price,product_id,quality,timestamp,uid

1,d8qShGJytuE3neTbo1N8M6HvOtyx9mXpfjB++YXk6uY=,200,14,DaywOe3qbtT3C8wBBSV+zBOH55DVz40L6PH1/1p9xCM=,14179,20.8,1,1,2017-05-25T14:34:24.079Z,11
1,d8qShGJytuE3neTbo1N8M6HvOtyx9mXpfjB++YXk6uY=,200,14,DaywOe3qbtT3C8wBBSV+zBOH55DVz40L6PH1/1p9xCM=,14179,20.8,1,1,2017-05-25T14:34:25.780Z,11
1,d8qShGJytuE3neTbo1N8M6HvOtyx9mXpfjB++YXk6uY=,200,14,DaywOe3qbtT3C8wBBSV+zBOH55DVz40L6PH1/1p9xCM=,14179,20.0,1,1,2017-05-25T14:34:30.887Z,11
1,d8qShGJytuE3neTbo1N8M6HvOtyx9mXpfjB++YXk6uY=,200,13,DaywOe3qbtT3C8wBBSV+zBOH55DVz40L6PH1/1p9xCM=,14179,20.0,1,1,2017-05-25T14:34:32.000Z,11

amount,	merchant_id,	price,	product_id,	quality,	timestamp
1,	A,	20.8,	1,	1,	2017-05-25T14:34:24.079Z
1,	A,	20.8,	1,	1,	2017-05-25T14:34:25.780Z
1,	A,	20.0,	1,	1,	2017-05-25T14:34:30.887Z
1,	A,	20.0,	1,	1,	2017-05-25T14:34:32.000Z

To Do: Data Preparation & Demand Learning

- Join relevant raw data (market situation & sales events)
- Dependent variable: amount, i.e, number of sales (match intervals!)
- Explanatory variables: Create suitable features from raw data
- Example, e.g., **merchant_id = A, product_id = 1,** $x_t^{(1)}(a,\vec{s}) = 1$:

	y_t	$x_t^{(2)}(a,\vec{s})$	$x_t^{(3)}(a,\vec{s})$
observation	**amount,**	**price_rank,**	**quality_rank, ...**
1	. . .		
2	. . .		
...			

- Apply logistic regression or other regression/ML techniques

Exercise

- Create similar raw data (market situations & sales) as csv-file
- We will also provide a platform-generated csv-file
- Join dependent and explanatory variables
- Apply logistic regression (e.g., via Python's scikit-learn)
- Exploit regression results to compute expected profits
  for certain time intervals and various market situations

Next Steps

- *Derive* a simple data-driven strategy (max short-term profits)
- Extension A: Price + *Quality* (Customer behavior is known)
- Extension B: Customer behavior is *not known* (Price, Price/Quality)
- Measure the long-term *performance* in a duopoly setting
- Evaluate your strategy against (un)known rule-based strategies

Mid-Term Goals & Presentation of First Results

- Presentations: First results (June 20)
- Describe your learning & pricing approach
- Describe your “pipeline”
- Present your performance results (in comparable setups)
- Apply *at least one alternative* regression / demand learning approach
- Present lessons learned & future work

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 & First Steps
8	**June 6**	**Workshop / Group Meetings**
9	June 13	Workshop / Group Meetings
10	June 20	Presentations (First Results)
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. :)