Data-Driven Decision-Making
In Enterprise Applications

Project Assignments

Rainer Schlosser

Hasso Plattner Institute (EPIC)

May 27, 2019

Today: Potential Projects

(1) Index Selection (LP)

(2) Data Placement for Replication (LP)

(3) Market Simulation + Demand Learning (DP)

(4) Duopoly Competition + Response Strategies (DP)

- Homework II: Dynamic Programming

Goals of the Project

- Understand & describe decision problem
- Derive solution approaches
- Apply learned optimization concepts & implement solution
- Simulate results & measure performance
- Presentation: Problem, approach, and early results
- Documentation: Summary of what has been done (until Aug 31)

(1) Index Selection

(1) Index Selection

Indexes can speed up the execution of queries.

But – indexes require memory and memory is limited.

Further, the impact of indexes is coupled:

The “best” indexes might not form the best selection!

What is “*index interaction*” (IIA)?

➡ The world’s best players do not form the best team!

(1) Index Selection – Problem Description

Context: Assume queries with different involved attributes (columns).

Suitable indexes can speed up queries, but require memory.

Decisions: Which *subset* of potential indexes to store?

Note, sets of index candidates and combinations are enormous

Impact: (i) What-if optimizer based costs *(no cost model!*)

(ii) Index interaction! (an index' utility is affected by others)

Constraints: Memory for indexes has a given limit (budget constraint)

Objective: *Minimize runtime s.t. the budget constraint*

(1) Index Selection – LP Formulation

Objective: minimize: Expected runtime (linear)

s.t.
- one index decision only for each query $j=1,\ldots,Q$
- index i used at all?
- budget constraint

Extensions: Stochastic workloads

Robust decisions

(2) Partial Replication

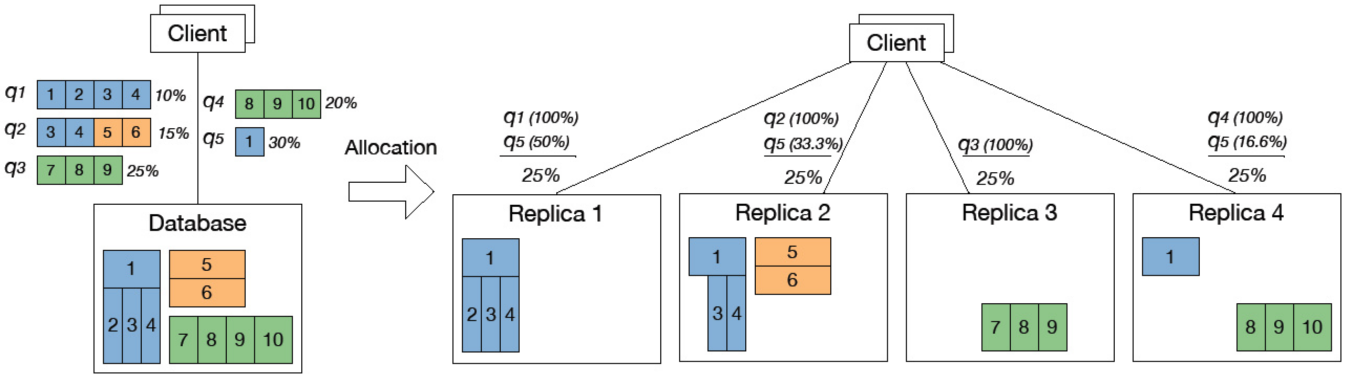
(2) Data Placement for Replication

If the workload exceeds a database’s capabilities *replicas* are used (scale-out).

We consider large *read-only* analytical workloads.

Workload can be distributed – but, storing data on replicas is costly!

How can we help the DBA to *balance* workloads with *minimal replicated* data?



(2) Data Placement for Replication

Context: Assume analytical read-only queries using different data fragments.

Workload is generated by queries (frequencies × costs).

Replica nodes take load from the master node.

Decisions: (i) which fragments to put on which replica (data placement)

(ii) which replica shall run which share of a query's workload

Impact: Deterministic

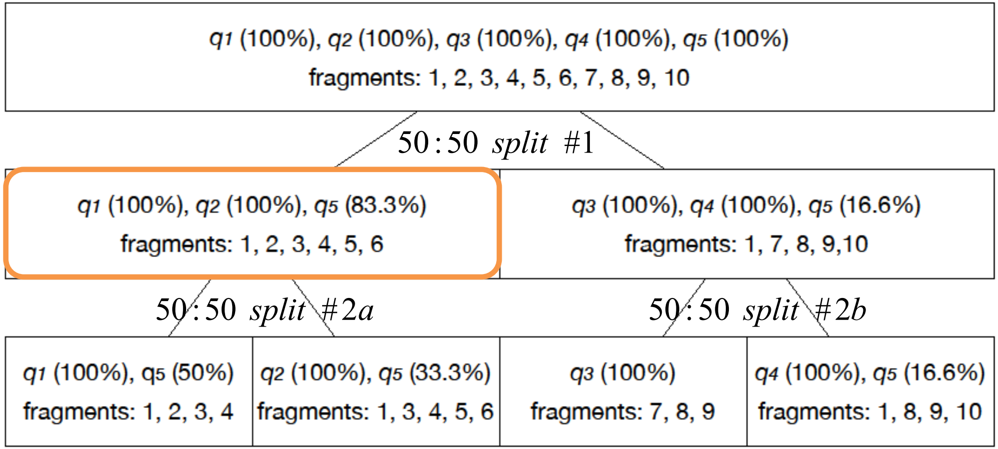
Constraints: (i) *Balance workload **evenly** on replicas*

(ii) To run a query on a replica *all* data fragments are needed

Objective: ***Minimize** costs of replicas (sum of required replicated data)*

(2) Solution Approach: LP-Based Decomposition

Optimization: LP-based decomposition (with scalable sub-problems)

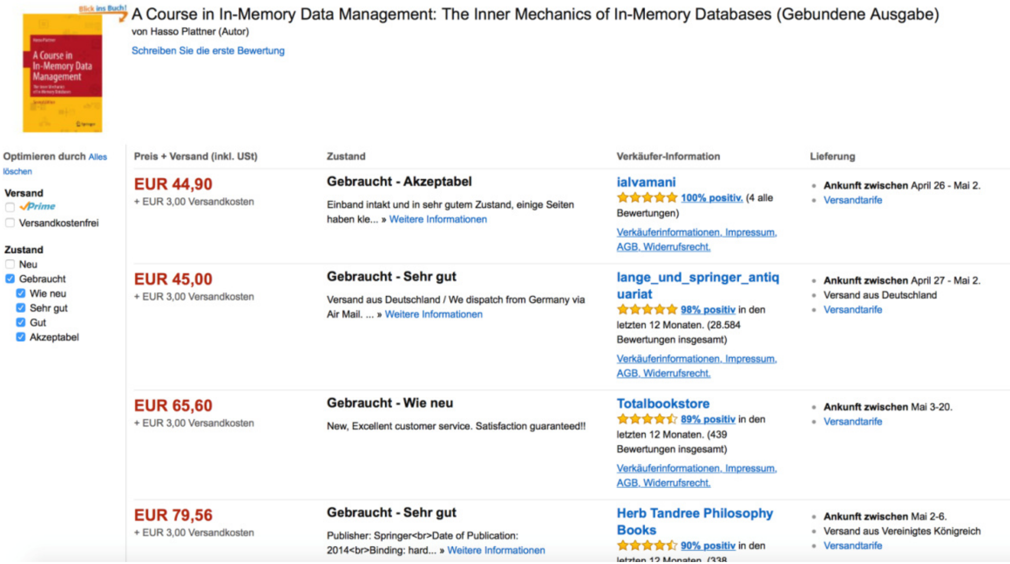


Extensions: Stochastic workloads

Robust decisions

(3) Markets & Demand Learning

(3) Market Simulation & Demand Learning



How can we assist an e-commerce merchant in optimizing his/her prices?

(3) Problem Description

Use-Case: A large merchant sells used books on Amazon Marketplace

Context:
(i) Many distinct items (ISBN), no reordering
(ii) Active competitors, changing environments
(iii) Multiple offer dimensions (quality, ratings, etc.)

Objective: Optimize expected profits & balance profitability vs. speed of sales

Decisions: Price updates

Impact: Effects of price updates **have to be estimated** from market data

Constraints: Limited inventory, limited price updates/hour

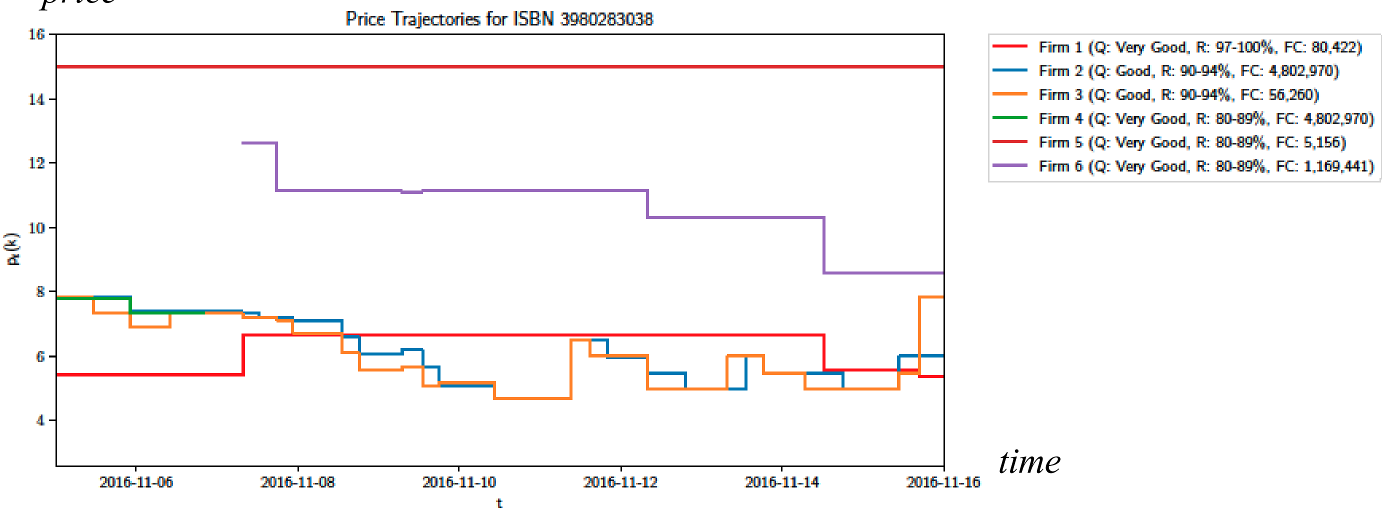
(3) Problem Description

Characteristics: – Exits & entries of competitors

– Active and passive competitors

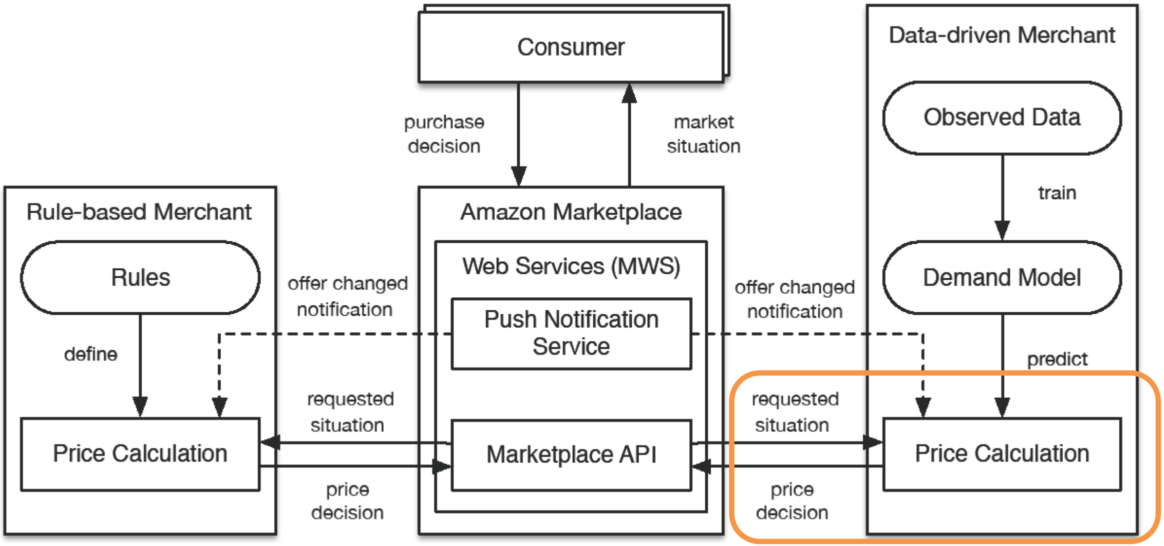
– Price cycles

price



(3) Process

- Price update process on Amazon: (i) request a market situation (ii) optimize price based on demand model, (iii) send price update



(3) Estimation of Demand and Optimization

(1) Estimation of Sales Probabilities

- ca. 10 market situations/day/item with 1-20 firms (100 Mio obs.)
- ca. 2 000 sales/month (1 year of data)
- Predict sales probabilities (for time intervals and market situations)

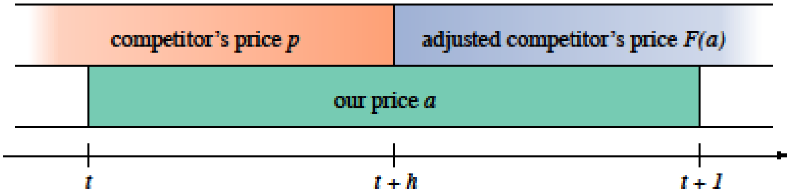
(2) Price *Optimization*

- Maximize expected discounted long-term profit
- Dynamic programming (with relaxed market anticipations)
- Computation time: should be fast

(4) Duopoly Competition

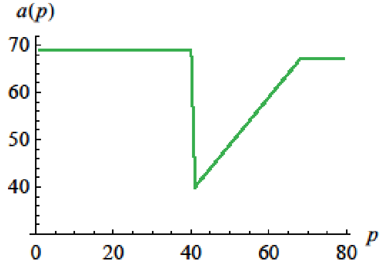
(4) Duopoly Competition & Response Strategies

Question: How do *optimal* price adjustment strategies look like?

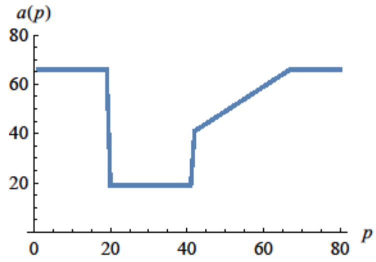


Setting: Infinite horizon, competitor's response strategy *is known*

Results:



against $F(a) := \max(a-1, 1)$

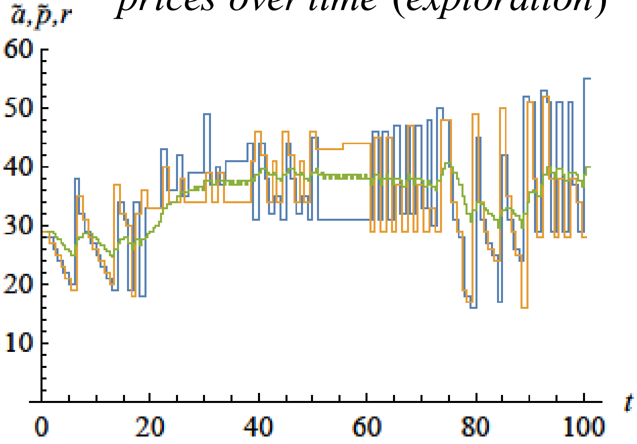


mutual optimal (equilibrium)

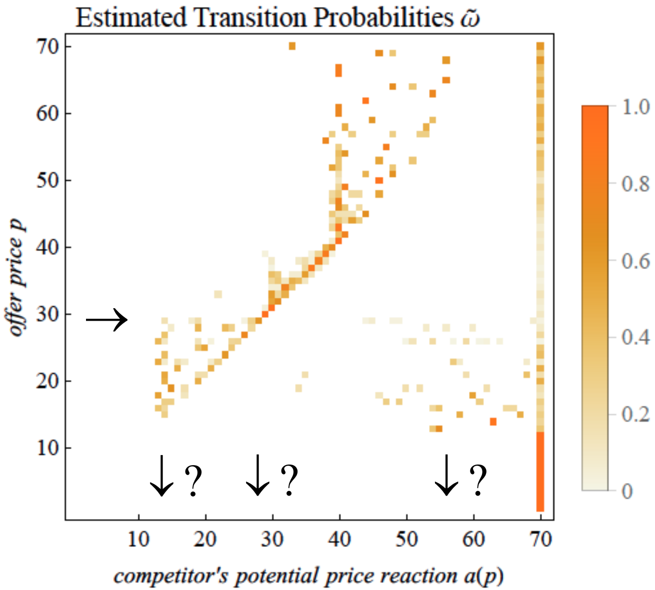
(4) Interaction of Self-Adapting Strategies (Short-Term)

- Now, price responses *have to be learned*!
- Both players update their strategies
- Do equilibria exist?

prices over time (exploration)



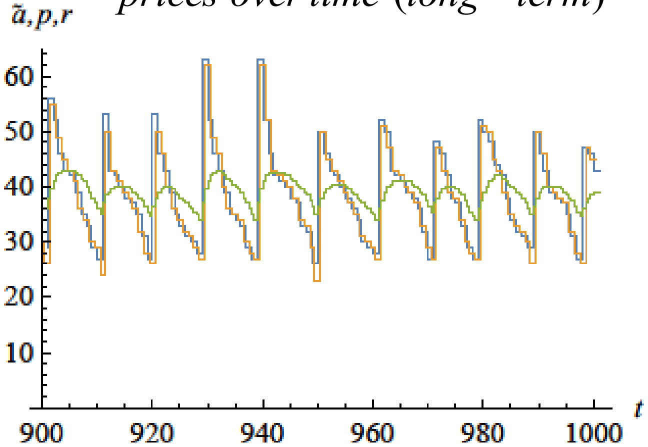
anticipated price reactions



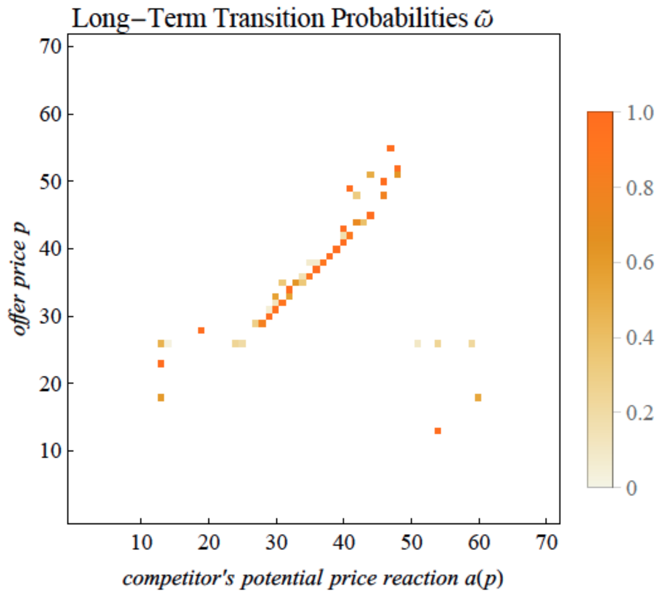
(4) Interaction of Self-Adapting Strategies (Long-Term)

- Now, price responses *have to be learned*!
- Both players update their strategies
- Equilibria in *mixed* strategies

prices over time $(long-term)$



long-term price reactions



Homework II

Homework II   Dynamic Programming

- A firm wants to sell airline tickets in a monopoly situation

- Setting:

  (i) N=100 initial items, T=100 periods time

  (ii) Known sales probabilities $P(i,a,t)$ (with #sales i, price a, period t)

  (iii) No terminal values, no overbooking, no cancellations

  (iii) Maximize expected total profit

- Evaluate expected and simulated results of the optimal policy

Homework II Sales Probabilities & Hints

- In period t: Binomial distr., $p(a,t) := (1 - a/200) \cdot t/T$, $t=0,\ldots,T-1$

$$P(i,a,t) := \binom{m}{i} \cdot p(a,t)^i \cdot \left(1 - p(a,t)\right)^{m-i}, \qquad i = 0,\ldots,m \quad a = 1,\ldots,200 \qquad m = 5$$

- Hint 1: Define Binomial coefficients recursively
- Hint 2: Choose decision in t for periods $(t,t+1)$, for all $t=0,\ldots,T-1$
- Hint 3: The state space is the number n of tickets left to sell at time t
- Hint 4: To determine the value function and its arg max,
  see example “Pricing Duopoly”, line 48-52

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	Pricing in Competitive Markets
8	May 27	Project Assignments + Homework 2 (until June 13)
9	**June 3**	**Robust Optimization Concepts**
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)