ORIE 5355

Lecture 10: Algorithmic pricing: price differentiation, competition, and practice

Nikhil Garg

Announcements & reminders

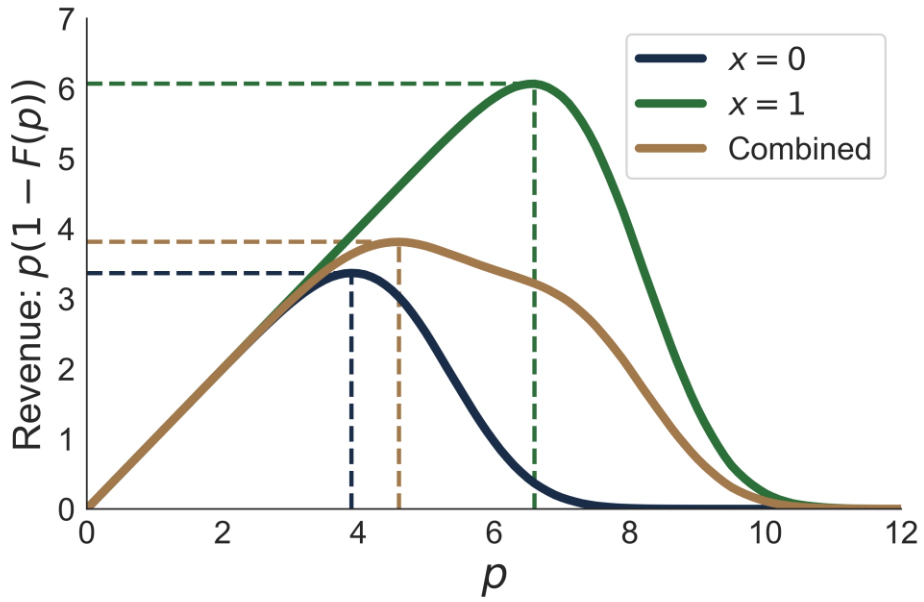
- Quiz 2 released – due Friday evening
- HW 3 released – due Tuesday 10/18
  - Conceptual component of HW due by class time on 10/05
  - Some changes made yesterday (very minor). Redownload please.

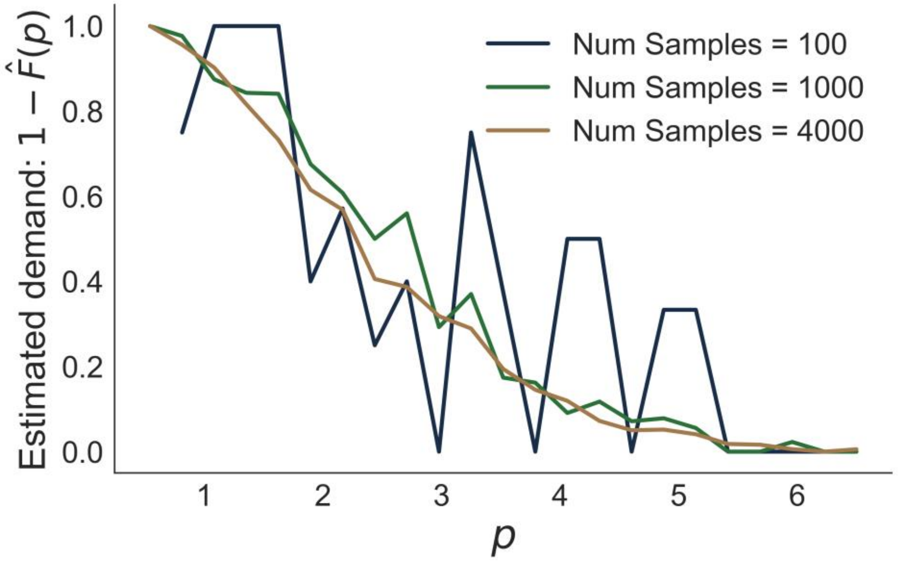
Pricing so far

- Given a demand distribution $d(p) = 1 - F(p)$, how to calculate optimal prices

$$\arg\max_{p} [p \times d(p)]$$

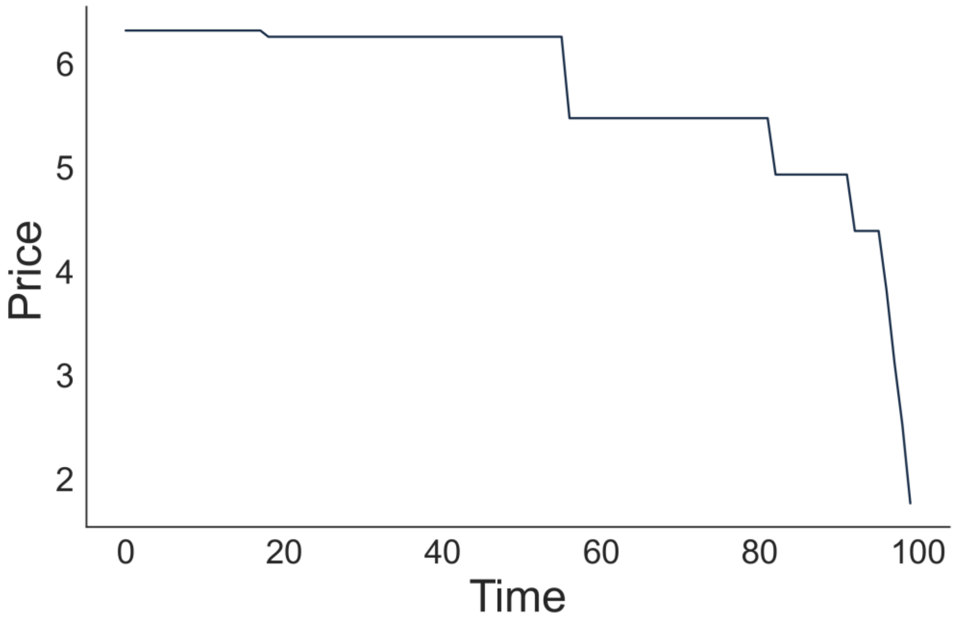
- How to estimate demand distributions, potentially as a function of covariates





Capacity constraints and pricing over time

- Dynamic programming approach
- If you have T time periods to sell an item and want to maximize expected revenue, what prices $p_1 \dots p_T$ do you set?
- Key idea: optimize backwards
  - First decide price p_T
  - Then decide price p_{T-1}
- Posted additional notes; come to OHs for additional questions



Plan for today

Last time:

- A little bit on using side-information (user and item vectors) to estimate personalized demand
- Capacity constraints over time

Many assumptions from previous lectures:

- Only one item
- Allowed to explicitly give different prices to different users
  - Or give different prices over time
- No competition from other sellers
- No over-time dynamics

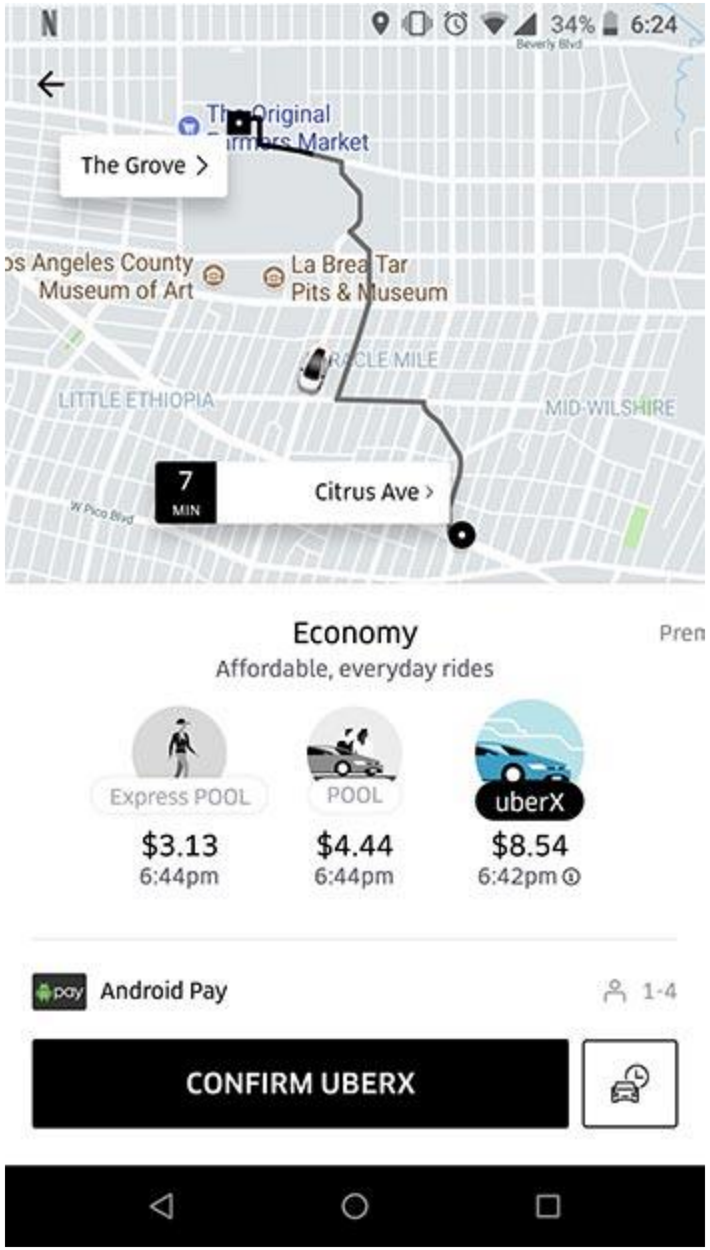
We’ll peel back some more of these assumptions today

Selling multiple kinds of items

Price differentiation

Example

- Ride-hailing offers different “tiers” of service
- UberPool cheaper than UberX
  - Also costs less for the platform
- How do we price these items together?
  - What happens if we do simple revenue maximizing price for each item separately?
  - What happens if we make UberPool cheaper?



Motivation

Motivation 1:

You simply have multiple kinds of products to sell. Different types of clothes, laptops, airline seats, furniture, etc.

Motivation 2:

- Earlier: personalized pricing with covariates
- Challenge: Often you can’t (technically, ethically, legally, …) give different prices for the same product to different users based on covariates
- Now: Different “tiers” of service.
  - High quality: First class seats, faster service in Uber/Lyft, luxury goods versions, get item “now”
  - Lower quality: Economy seats, UberPool/Lyft Wait and Save, …

=> Purposely create tiers of service to earn more money from richer people while earning something from others

Challenges

- Just like pricing over time, now prices for the 2 items depend on each other

  Unlike pricing to different demographic segments without capacity constraints

- Cannibalization: Customers who would have bought the luxury good instead buy the cheaper good because it is available

2-item user behavior model

- Suppose you're selling 2 types of items
- Each person will buy at most one item
  - Each person has a *private valuation* v_1 for item 1 and v_2 for item 2
- Suppose you offer the items at price p_1 and p_2, respectively
- How does the person make their decision?

  Utility from item j at price p_j is $v_j - p_j$
- Person i buys

  Neither item if $v_1 < p_1$ and $v_2 < p_2$
  Item 1 if $v_1 \geq p_1$ and $v_1 - p_1 \geq v_2 - p_2$
  Item 2 if $v_2 \geq p_2$ and $v_2 - p_2 \geq v_1 - p_1$

Assumption on customer's "choice model." More generally, customer could buy randomly, with choice probabilities that depend on

$$v_j - p_j$$

In more detail

How does the person make their decision? Person i buys

- Neither item if $v_1 < p_1$ and $v_2 < p_2$
- Item 1 if $v_1 \geq p_1$ and $v_1 - p_1 \geq v_2 - p_2$
- Item 2 if $v_2 \geq p_2$ and $v_2 - p_2 \geq v_1 - p_1$

Revenue in 2 item model

For a set of prices (p_1, p_2), let

$\mathrm{d}_1(p_1, p_2)$ be fraction of people who buy item 1

(Yellow Region)

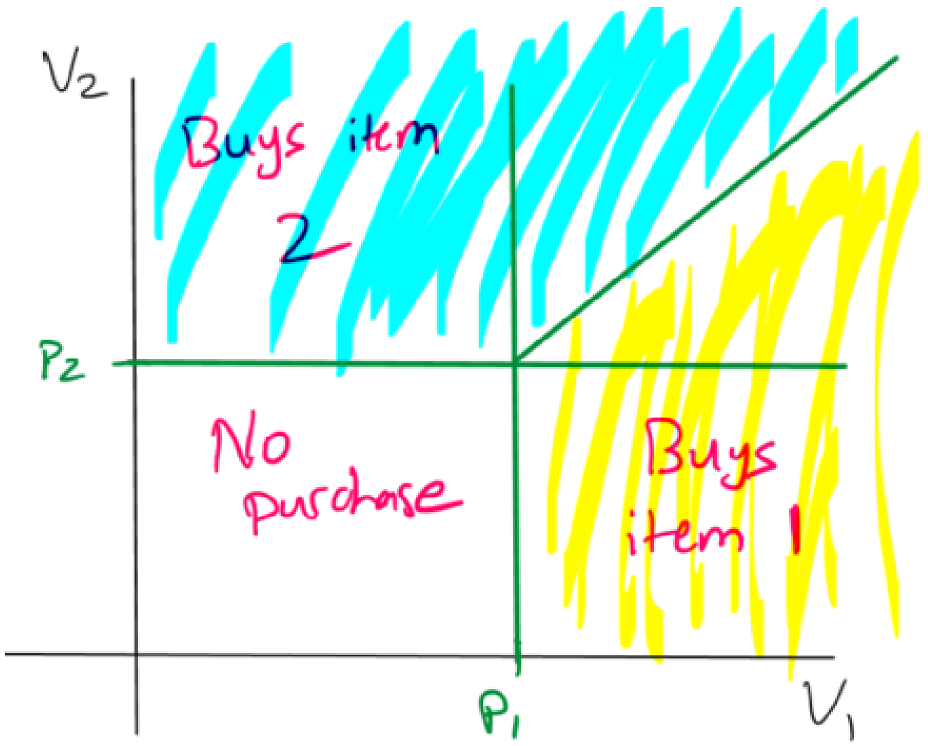
$\mathrm{d}_2(p_1, p_2)$ be fraction of people who buy item 2

(Blue Region)

Then, revenue is:

$$p_1 \times \mathrm{d}_1(p_1, p_2) + \ p_2 \times \mathrm{d}_2(p_1, p_2)$$

Given functions $\mathrm{d}_1, \mathrm{d}_2$, can solve for optimal prices



Cannibalization

Now, each price affects other item.

Revenue: $p_1 \times \mathrm{d}_1(p_1, p_2) + \; p_2 \times \mathrm{d}_2(p_1, p_2)$

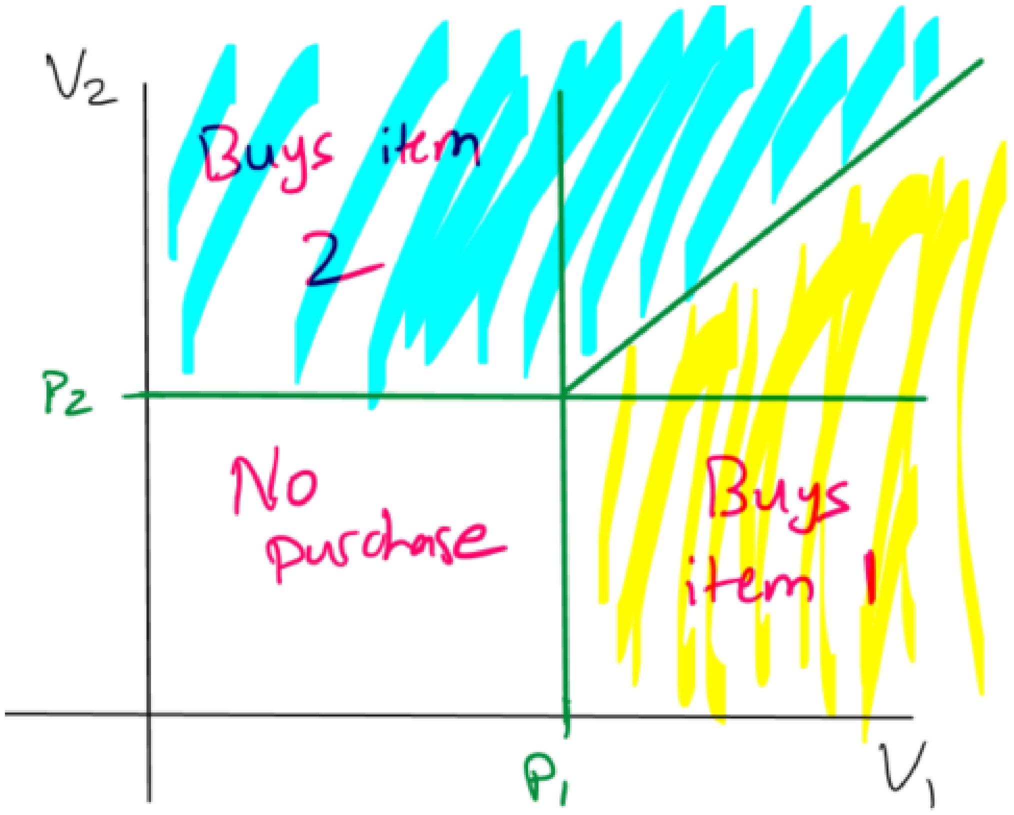
Suppose decrease p_1 (make item 1 cheaper)

Then:

- Earn less money in yellow region ↓
- Yellow region becomes bigger

    White region becomes smaller ↑

    Blue region becomes smaller ↓



Demand estimation with multiple items

- With a single item, we suggested machine learning approach to estimate: $\mathrm{d}(\mathrm{p}, \mathrm{x}) \stackrel{\text{def}}{=} 1 - F_{p|X}\,(p \mid X = x)$
- Assume we have user i with covariates x_i
- Now, would need to estimate $\mathrm{d}_1(p_1, p_2, x_i)$ and $\mathrm{d}_2(p_1, p_2, x_i)$
- Gets very hard, very quickly
- Approach 1: Use a *multi-class* classification algorithm $g(p_1, p_2, x_i)$
  [Buy nothing, buy item 1, buy item 2] and then extract class probabilities
  (sci-kit learn: use **predict_proba** with any multi-class classifier)
- Approach 2: (Extend idea from previous class)
  - Use user and item vectors, i.e., $(\mathrm{p}_1, \mathrm{p}_2, u_i \cdot w_{\text{item }1}, u_i \cdot w_{\text{item }2})$

Sidenote: Substitutes and complements

- So far: motivation -- we have multiple products to sell, that appeal to different customers
  "cheaper" and "more expensive" product
- Items are "substitutes": people only buy at most one kind of item
- Sometimes, items are "complements" – buying one item makes the other item more attractive
  - Soda + popcorn at movie theater
  - iPhone and Macbook and Apple Watch and Apple TV and …
- Then, reducing one item's price might induce you to buy more overall
  - An item is a "loss leader"

Putting pieces together: class competition

So far we’ve covered

- Recommendation systems
  - Given past user and item data, predicting how much each user would like each item
  - How to turn these predictions into *recommendations* (with capacity constraints)
- Pricing
  - Single item revenue maximization
  - Estimating demand at each price, potentially with covariates
    - Potentially with multiple items, and with using user and item vectors
  - Pricing over time with capacity constraints
  - Pricing multiple items

Overview: Real-life algorithmic pricing

- You and a single competitor (your classmates) each are selling two types of items, Book A and Book B.
  - ~~With some initial capacity of each (let's pretend 10) No capacity constraints~~
- A customer walks in and you observe some personal data
  - Just demographic covariates
  - Demographic covariates & user vector trained using their past experiences
- You and your competitor post prices for each item
- The customer at most 1 item and leaves
- Repeat for many customers over time

Basic case

- For now, let's ignore: Competition and capacity constraints
- For each user, you have either just demographic covariates x_i or also a trained user vector u_i from their past interactions on your site
- You would predict demand for each item, $d_1(p_A, p_B\ x_i, u_i)$ and $d_2(p_A, p_B, x_i, u_i)$ for each set of prices (p_A, p_B)
  - Your choice on how to estimate this demand
  - What do you do for customers with no user vector u_i?
- Set prices to maximize your expected revenue

Complication 1: Capacity constraints

- ~~Now, have 10 copies of each item, and there will be T=100 customers.~~
- ~~Now, the price that you set for each item should depend on opportunity cost: what if you can sell that item to a different customer in the future?~~
- ~~3-d Bellman equation: time, capacity of Book A, capacity of Book B~~
- ~~Set up your Bellman equation:~~

$$V_{t,k_A,k_B} = A + B + C$$

~~A: If I sell Book A today: Revenue today, plus future revenue from 1 less Book A~~

~~B: If I sell Book B today : Revenue today, plus future revenue from 1 less Book B~~

~~C: If I don’t sell anything: future revenue from same number of copies~~

How to calculate future revenue?

- ~~As before, future revenue depends on future prices that you set~~
- ~~…Think about prices you'd set on last day T-1=99~~
  - ~~For each combination of capacities left k_A, k_B~~
- ~~Complication: on day $t < T - 1$ you don't yet know the customer x_{T-1}, u_{T-1} that will show up on the last day $T - 1$~~!
  - ~~You only know customer who has shown up on day t~~
- ~~When calculating future *expected* value V_{t+1,k_A,k_B}, you need to consider the *distribution* of customers that *could* show up~~
  - ~~Use training data to consider possible customers that could show up~~
  - ~~Then calculate the prices that you *would* show each of them~~

Complication 2: Competition

- You and your opponent both do the same thing, and calculate the exact same prices p_A, p_B at the current time step
- Your opponent is clever, and so decides to *undercut* you slightly, and so sets prices $p_A - \$0.01, p_B - \$0.01$
- …but you're cleverer, and know your opponent will do this, and so you set prices $p_A - \$0.02, p_B - \$0.02$
- There's now a game theory component: you need to anticipate what your opponent will do when setting prices
- More complicated: it's a repeated setting
  - If you "lose" today, your competitor has less items in stock for tomorrow
  - You can *learn parameters* for how your opponent behaves

Rest of pricing module

10/3: Pricing in ride-hailing

10/5: What's acceptable in pricing?

- **Required to complete the questionnaire before the class!**

Questions?