ORIE 5355

Project description, game theory/competition

Nikhil Garg

Plan for rest of the semester

Almost there!

Rest of semester

- Class project
- Miscellaneous lectures, guest lectures
  - Quiz 5
  - Attendance

Project Overview

Deadlines

Official deadlines

- Part 1 – November 21
- Part 2 – December 3 [BUT DO NOT WAIT TO START]
- Report – December 12

Can submit edits to Part 1 + Part 2 at report deadline if absolutely needed (substantial points deduction)

Earlier "deadlines"

Starting this week, will start running preliminary Part 1 tournaments

- Such earlier submissions highly recommended for debugging purposes!

Will also start running preliminary Part 2 tournaments soon

Part 1 overview

- You and a single competitor (your classmates) each are selling one identical item, Book A.
- A customer walks in and you magically see their valuation for item v
- You and your competitor post prices for each item
- The customer buys from one of you, or neither
  - If minimum price < v, buys from minimum price seller
- Repeat for many more customers
- Objective: maximize revenue across games (1 game against each team)

Deliverable: Code for competitive pricing

Part 2 overview

Changes from Part 1:

- Now are selling 2 items, Book A and Book B.
- Observe customer **covariates**

=> Even without competition, must do demand estimation + price optimization

Deliverables

- Non-competitive demand estimation + price optimization:
  - For a set of test customers, what prices would you set for each customer for each item to maximize revenue, *if you didn't have to worry about competition?*
- Code for competitive pricing

Grading

Project report	0-6 points	How clearly + well did you *discuss* your strategy and evaluation techniques? How thoughtful was your reflection?
Project Part 1	0-6 points	How thoughtful and effective was your strategy? Does it reflect substantial effort? This will be influenced by objective performance, description in your report, and looking at your code.
Project Part 2	0-6 points	Same as Part 1 above
Overall subjective	0-2 points	

For grades: performance matters, but *strategy/analysis* matters more

Code Submission information

- Via GitHub classroom!
- Github classroom has an autograder that checks file structure and packages!

Summary

- Fill out the google form with your team's information. At least 1 team member needs GitHub username
- Have 1 team member click on github classroom invite link and "create" the team (use same team name as form)
  - This creates a repository
  - Everyone else can click on same link and join the team via team name.
- Rename "yourteamname.py" agent file
  - And rename "yourteamname/" folder. Allowed to put data/pickled machine learning models/etc into this folder
- Submission of code is just via pushing to the repository

How to avoid common submission mistakes

- Do **not** edit files other “yourteamname.py” or “yourteamname/” (or at least, do not rely on those edits for your code to work)
- Do **not** edit **function names** inside “yourteamname.py” file
- **Yes:** Edit above **file names** “yourteamname.py”, “yourteamname/” with your team name
- Only put your required data or machine learning models in the folder “yourteamname/” inside the agents folder
- **Yes:** Use *relative* paths when loading data in your agent

  CORRECT: `filename = 'agents/yourteamname/trained_model’`

  NOT CORRECT: `filename = ‘c:/Users/Nikhil/documents/ORIE5355bestclassever/project/agents/yourteamname/trained_model’`

Part 1 suggestions + brief intro to game theory

More broadly: game theory, pricing competition

Challenge with competition

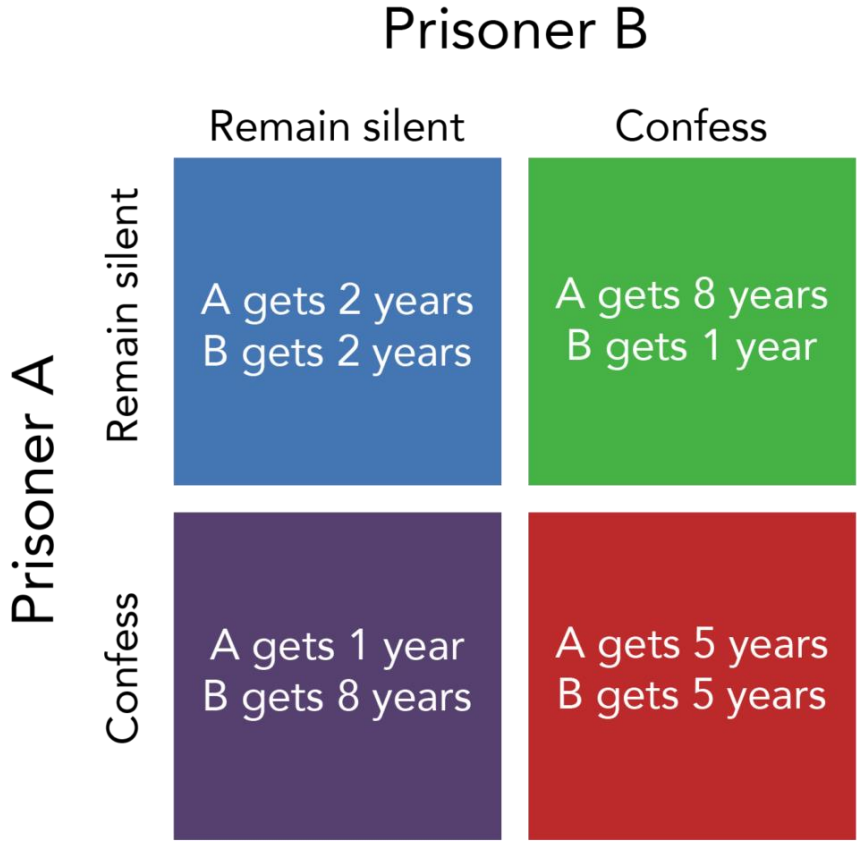
- There's now a game theory component: you need to anticipate what your opponent will do when setting prices
- You and your opponent both do the same thing, and calculate the exact same price p at the current time step
- Your opponent is clever, and so decides to *undercut* you slightly, and so sets price $p - \$0.01$
- …but you're cleverer, and know your opponent will do this, and so you set prices $p - \$0.02$
- … You keep doing this, and you're both setting prices $p = 0$
- You can't ignore your opponent, otherwise they'll undercut you and always win the customer

Aside: game theory

Game theory: "study of mathematical models of strategic interactions among rational agents"

Example: Prisoner's dilemma

- 2 players
- Each can take 1 of 2 actions
- Have to choose action without knowing what other person chose
- **No matter what other person does**, you're better off confessing



[Prisoner's Dilemma | Microeconomics (lumenlearning.com)]

What about *repeated* games?

- In many settings, you're playing a *repeated* game
- Sometimes, repeated games change optimal behavior:
  - If I betray you today, you can retaliate by betraying me tomorrow
  - If I remain cooperate today, you can reward me tomorrow
- …but, calculating optimal strategies in repeated settings can get complicated
- There are tournaments where people just play repeated prisoner's dilemma against each other. Person with most utility overall wins
  - "tit-for-tat" is often a top performing strategy

Pricing in repeated settings

- By the logic of a previous slide, setting prices $p = 0$ is only rational response if you want to win the customer
- BUT, your objective isn't to win the customer, it's to make the most money overall. (and you know your opponent's goal is the same)
- What you do should depend on your opponent's actions
  - If they tend to price low, you should also price low
  - If they tend to price high, you should also price high
- But – your actions affect your opponent's future actions!

  How do you avoid a price war?

Part 2: 2 items, user covariates

Step 1: Demand estimation + price optimization (HW 3…again)

Demand estimation

- Given buyer covariates and (single seller) prices p_A, p_B: what is the probability that the buyer buys item A? Item B? neither item?
- Task: train a machine learning model given the training data
  - (Data is now in the repository)

Price optimization

- Given your demand estimation, what are optimal prices p_A, p_B? (to maximize revenue if you didn't have competition?)
- Now, have to do "2-dimensional" optimization.

Deliverable: optimal non-competitive prices for a set of customers

Step 2: Competition

Put Project Part 1 together with demand estimation/price optimization:

How to adjust your estimated prices over time against each opponent?

Questions?