PhD Industrial Organization: Homework 1

Due: See Canvas.
You can work in groups of up to 2 people.

Instructions

- You can work in groups of 2 people. It might be a good idea to code up the solutions individually and then compare answers so that you both get experience with the code and understand the solutions.

- As you work through the questions, I strongly recommend using Git to version control your code (and pushing to your Github account as you make changes). Git is also great for collaboration if you are working in a group of 2.

- Please hand in both your solutions and also the code you used to generate the solutions. You can work in whichever programming language you want.

- This assignment is based on one written by Allan Collard-Wexler.

- Good luck!

Background and Data

In this homework assignment you will be coding up the BLP algorithm (amongst other things). You will be applying demand estimation to a famous 'fake cereal dataset' which used to be found on Aviv Nevo's website, and which I downloaded from the PyBLP tutorial module (by Conlon and Gortmaker (2020)) and modified slightly. The datasets are synthetic - presumably the original cereal data could not be shared publicly - but retain some of the properties of the original dataset and so can be used for estimation.

There are two datasets:

- **product_data.csv**

- This contains data about market shares, prices, and product characteristics.

- The descriptions of the variables are:

    - *market_ids*: IDs for the different markets. The ID denotes a city-quarter. So, 'C01Q1' captures a city (e.g. 'C01') in a quarter ('Q1').

- *product_ids*: IDs for the different products. A single code might correspond to, for example, 'Apple-Cinnamon Cheerios'

- *shares*: market shares (precomputed)

- *prices*: prices of a cereal box

- *sugar*: amount of sugar

- *demand_instruments0, ..., demand_instruments19*: precomputed demand instruments (these were the ones used in the original paper).

- **agent_data.csv**

- This contains 20 precomputed random draws for each market which you can use to do monte-carlo integration. (Note that if you were doing this in a research paper you should probably use more draws.)

- The descriptions of the variables are:

  - *market_ids*: IDs for the different markets. The ID denotes a city-quarter. So, 'C01Q1' captures a city (e.g. 'C01') in a quarter ('Q1').

  - *weights*: weight of each draw ($= 1/20$)

  - *income*: draw from the distribution of income in each market (this is an observed demographic)

  - *nodes0*: draw from a standard normal (this is used in the computation of the unbserved taste for characteristics)

Good resources if you have questions

- It is extremely likely that you will run into some confusion about how all the pieces in the algorithm fit together. After all, understanding this is the point of the assignment! Here are some good places to search if you have questions:

- Conlon and Gortmaker "Best Practices For Differentiated Products Demand Estimation With PyBLP" (2020, Rand): this paper clearly sets out the algorithm, tricks that people have found useful to speed up the computation, and what works and what doesn't.

- The corresponding PyBLP Python Module: docs at $https://pyblp.readthedocs.io/en/stable/$. This is where I pulled the data from. I kept the column names the same in case you want to use the results of PyBLP as a 'source of truth' to check that your algorithm is working properly.

- You can also talk to me after class, send me an email to set up a meeting, or just ask the question by email: nvreugde@asu.edu.

1. Logit Model

Consider the homogeneous-consumer logit model:

$$u_{ijt} = x_{jt}\beta + \alpha p_{jt} + \xi_{jt} + \epsilon_{ijt}$$

Here the notation is the same as in Handbook Chapter 2 and the lecture slides. Specifically:

- u_{ijt}: utility of consumer i of product j in market t

- x_{jt}: observed characteristics of product j in market t (note: this should <u>include a constant</u> as well as the variable 'sugar')

- p_{jt}: price of product j in market t

- ξ_{jt}: unobserved demand shifter

- ϵ_{ijt}: i.i.d. logit draws

Instructions:

a. Estimate the above model using OLS and report the coefficients (no need to report standard errors).

b. Estimate the above model using 2SLS, instrumenting for price using the instruments: *demand_instruments0, ..., demand_instruments19*. Report the coefficients (no need to report standard errors).

c. Using your estimated coefficients from Part (b.) compute own-price elasticities for each product in the market 'C01Q1'. Draw a scatterplot with the computed own-price elasticities on the y-axis and prices on the x-axis (each dot on the scatterplot should correspond to an individual product in the market 'C01Q1'). Discuss the observed relationship between prices and the own-price elasticities predicted by the model.

2. Mixed Logit/ BLP Model

Consider the mixed logit/BLP model:

$$u_{ijt} = x_{jt}\beta_{it} + \alpha_{it}p_{jt} + \xi_{jt} + \epsilon_{ijt}$$

In the above equation the notation is the same as in Handbook Chapter 2 and the lecture slides. The coefficients that comprise the vector β_{it} are a function of income but do not have a random component and are given by:

- $\beta_{it}^0 = \beta_0^0 + \beta_{income}^0 [income_{it}]$

- $\beta_{it}^{sugar} = \beta_{income}^{sugar} [income_{it}]$

The coefficient on price is a function of the consumer's income and a random part (denoted by ν_{it} where ν_{it} is drawn from a standard normal distribution):

- $\alpha_{it} = \alpha_0 + \alpha_{income} [income_{it}] + \alpha_\nu \nu_{it}$

Therefore, there are six unknown parameters. There are two 'linear' parameters: β_0^0 and α_0. There are four 'nonlinear' parameters: $\beta_{income}^0, \beta_{income}^{sugar}, \alpha_{income}, \alpha_\nu.$[1]

Instructions:

a. Estimate the above model using the BLP method and report the parameters (please write your own code; no need to report standard errors). To build the moments in the GMM objective function include the product characteristic 'sugar' as an instrument as well the precomputed demand instruments. For the weighting matrix choose the 'optimal weight matrix' $W = (Z'Z)^{-1}$ where Z is the vector of instruments.

b. Using your estimated parameters from Part (a.) compute own-price elasticities for each product in the market 'C01Q1'. Draw a scatterplot with the computed own-price elasticities on the y-axis and prices on the x-axis (each dot on the scatterplot should correspond to an individual product in the market 'C01Q1'). Discuss the observed relationship between prices and the own-price elasticities predicted by the model. Also, discuss differences with Part 1(c.)

c. (*Optional/challenge question*) Using the ASU Sol cluster, submit a job which computes the model optimization at 10 different initial starting points. To choose the initial points, pick one of the parameters to vary (say, α_{income}) and fix the other parameters. The batch job should itself submit 10 sub-jobs, with each job running the optimization for one of the starting points (for more information, search for *'SLURM job array'* or see here).[2] Does the model optimization find the same coefficients regardless of the starting point, or does it hit local minima?

---

[1]Presumably you could write down a more flexible model allowing for more interactions etc - the above model is deliberately simplified to make the computation easier.

[2]This exercise closely mirrors the kinds of things you might use the ASU Sol cluster for if you were using a structural model in your research. For example, you might use a batch job if you were bootstrapping your standard errors. An extra benefit of attempting this challenge question is that you will probably be able to repurpose most of your submission script in future modeling work.