PhD Empirical IO
Fall 2019
Prof. Conlon
Homework Assignment[1]
Due Oct 22

Overview

You will estimate demand and supply in a stylized model of the market for pay-TV services. You will use any programming language (Python/R/Matlab/Julia) to create your own fake data set for the industry and do some relatively simple estimation. Then, using the `pyBLP` package of Conlon and Gortmaker, you will estimate the model and perform some merger simulations.[2]. The pyBLP package has excellent documentation and a very helpful tutorial (which covers merger simulation), both easy to find (`https://pyblp.readthedocs.io/en/stable/`). You may want to work through the tutorial notebooks available with the documentation (or on the Github page).

To install `pyBLP` you need to have Python 3 installed, I recommend Anaconda `https://www.anaconda.com/distribution/`. If you have python installed you simply need to type:

```
pip install pyblp
```

or

```
pip install git+https://github.com/jeffgortmaker/pyblp
```

Please submit a single printed document presenting your answers to the questions below, requested results, and code. Write this up cleanly with nice tables where appropriate. You may work in groups of up to 3 on the coding, but your write-up must be your own work and must indicate who your partners are.

You can do parts (2) and (3) in R, Matlab, Julia, or Python. Parts (4) and (5) use `pyblp` which you can run in R using `reticulate` if you really want.

1 Model

There are T markets, each with four inside goods $j \in \{1,2,3,4\}$ and an outside option. Goods 1 and 2 are satellite television services (e.g., DirecTV and Dish); goods 3 and 4 are wired television services (e.g., Frontier and Comcast in New Haven). The conditional indirect utility of consumer i for good j in market t is given by

$$\begin{aligned} u_{ijt} &= \beta^{(1)} x_{jt} + \beta_i^{(2)} satellite_{jt} + \beta_i^{(3)} wired_{jt} + \alpha p_{jt} + \xi_{jt} + \epsilon_{ijt} \qquad j > 0 \\ u_{i0t} &= \epsilon_{i0t}, \end{aligned}$$

[1] Thanks to Phil Haile and Jaewon Lee for coding tips and other highly useful feedback on this problem set.

[2] Using data you generate yourself gives you a way to check whether the estimation is working; this is a good thing to try whenever you code up an estimator!

where x_{jt} is a measure of good j's quality, p_{jt} is its price, $satellite_{jt}$ is an indicator equal to 1 for the two satellite services, and $wired_{jt}$ is an indicator equal to 1 for the two wired services. The remaining notation is as usual in the class notes, including the i.i.d. type-1 extreme value ϵ_{ijt}. Each consumer purchases the good giving them the highest conditional indirect utility.

Goods are produced by single-product firms. Firm j's (log) marginal cost in market t is

$$\ln mc_{jt} = \gamma^0 + \mathrm{w}_{jt}\gamma^1 + \omega_{jt}/8,$$

where w_{jt} is an observed cost shifter. Firms compete by simultaneously choosing prices in each market under complete information. Firm j has profit

$$\pi_{jt} = \max_{p_{jt}} M_t(p_{jt} - mc_{jt})s_{jt}(p_t).$$

2 Generate Fake Data

Feel free to use the software package of your choice

Generate a data set from the model above. Let

$$\begin{aligned} \beta^{(1)} &= 1,\ \beta_i^{(k)} \sim \text{iid } N(4,1) \ \text{ for } k = 2,3 \\ \alpha &= -2 \\ \gamma^{(0)} &= 1/2,\ \gamma^{(1)} = 1/4. \end{aligned}$$

1. Draw the exogenous product characteristic x_{jt} for $T = 600$ geographically defined markets (e.g., cities). Assume each x_{jt} is equal to the absolute value of an iid standard normal draw, as is each w_{jt}. Simulate demand and cost unobservables as well, specifying

$$\begin{pmatrix} \xi_{jt} \\ \omega_{jt} \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 0.25 \\ 0.25 & 1 \end{pmatrix} \right) \text{ iid across } j, t.$$

2. Solve for the equilibrium prices for each good in each market.

   (a) Start by writing a procedure to approximate the derivatives of market shares with respect to prices (taking prices, shares, x, and demand parameters as inputs). The key steps are:

      i. For each (j, t), write the choice probability s_{jt} as a weighted average (integral) of the (multinomial logit) choice probabilities conditional on the value of each consumer's random coefficients;

      ii. Anticipating differentiation under the integral sign, derive the analytical expression for the derivative of the *integrand* with respect to each p_{kt};

      iii. Use the expression you obtained in (2) and simulation draws of the random coefficients to approximate the integral that corresponds to $\partial s_{jt}/\partial p_{kt}$ for each j and k (i.e., replace the integral with the mean over the values at each simulation draw).

      iv. Experiment to see how many simulation draws you need to get precise approximations and check this again at the equilibrium shares and prices you obtain below.

Note: you do not want to take new simulation draws of the random coefficients each time you call this procedure. This is because, if you did so, the attempt to solve for equilibrium prices (below) may never converge due to "jittering" across iterations. So take your simulation draws only once, outside the procedure you write here.

(b) The FOC for firm j's profit maximization problem in market t is

$$(p_{jt} - mc_{jt})\frac{\partial s_{jt}(p_t)}{\partial p_{jt}} + s_{jt} = 0$$

$$\implies p_{jt} - mc_{jt} = -\left(\frac{\partial s_{jt}(p_t)}{\partial p_{jt}}\right)^{-1} s_{jt} \tag{1}$$

(c) Substituting in your approximation of each $\left(\frac{\partial s_{jt}(p_t)}{\partial p_{jt}}\right)$, solve the system of equations (**??**) (J equations per market) for the equilibrium prices in each market.

i. To do this you will need to solve a system of $J \times J$ nonlinear equations.[3] Make sure to check the exit flag for each market to make sure you have a solution.

ii. Do this again using the algorithm of Morrow and Skerlos (2011), discussed in section 3.6 of Conlon and Gortmaker (2019) (and in the `pyBLP` "problem simulation tutorial"). Use the numerical integration approach you used in step (a) to approximate the terms defined in equation (25) of Conlon and Gortmaker. If you get different results using this method, resolve this discrepancy either by correcting your code or explaining why your preferred method is the one to be trusted.

3. Calculate "observed" shares for your fake data set using your parameters, your draws of x,w, β_i, ω, ξ, and your equilibrium prices.

3 Estimate Some Mis-specified Models

Feel free to use the software package of your choice

4. Estimate the plain multinomial logit model of demand by OLS (ignoring the endogeneity of prices).

5. Re-estimate the multinomial logit model of demand by two-stage least squares, instrumenting for prices with the exogenous demand shifters x and excluded cost shifters w. Discuss how the results differ from those obtained by OLS.

6. Now estimate a nested logit model by two-stage least squares, treating "satellite" and "wired" as the two nests for the inside goods. You will probably want to review the discussion of the nested logit in Berry (1994). Note that Berry focuses on the special case in which all the "nesting parameters" are the same; you should allow a different nesting parameter for each nest.[4] Without reference to the results, explain the way(s) that this model is misspecified. (Hint: students tend to get this question wrong; recall that I suggested you review Berry 94).

[3] To do this in Python use `scipy.optimize.root`, in MATLAB use `fsolve`, in R use `nleqslv` or something similar, and in Julia use `NLSolve.jl` or something similar.

[4] In Berry's notation, this means letting the parameter σ become $\sigma_{g(j)}$, where $g\,(j)$ indicates the group (satellite or wired) to which each inside good j belongs.

7. Using the nested logit results, provide a table comparing the estimated own-price elasticities to the true own-price elasticities.[5] Provide two additional tables showing the true matrix of diversion ratios and the diversion ratios implied by your estimates.[6]

4 Estimate the Correctly Specified Model

Use the `pyBLP` package to estimate the correctly specified model. Allow `pyBLP` to construct approximations to the optimal instruments, using the exogenous demand shifters and exogenous cost shifters.[7]

8. Report a table with the estimates of the demand parameters and standard errors. Do this three times: once when you estimate demand alone, then again when you estimate jointly with supply; and again with the "optimal IV".

9. Using your preferred estimates from the prior step (explain your preference), provide a table comparing the estimated own-price elasticities to the true own-price elasticities. Provide two additional tables showing the true matrix of diversion ratios and the diversion ratios implied by your estimates.

9* Extra Credit. Bootstrap your diversion ratio estimates and compare the bootstrapped confidence interval to the "true" values. (This may take some computer time).

5 Merger Simulation

10. Suppose two of the four firms were to merge. Give a brief intuition for what theory tells us is likely to happen to the equilibrium prices of each good j.

11. Suppose firms 1 and 2 are proposing to merge. Use the `pyBLP` merger simulation procedure to provide a prediction of the post-merger equilibrium prices.

12. Now suppose instead that firms 1 and 3 are the ones to merge. Re-run the merger simulation. Provide a table comparing the (average across markets) predicted merger-induced price changes for this merger and that in part 11. Interpret the differences between the predictions for the two mergers.

13. Thus far you have assumed that there are no "efficiencies" (reduction in costs) resulting from the merger. Explain briefly why a merger-specific reduction in marginal cost could mean that a merger is welfare-enhancing.

14. Consider the merger between firms 1 and 2, and suppose the firms demonstrate that by merging they would reduce marginal cost of each of their products by 15%. Furthermore, suppose that they demonstrate that this cost reduction could not be achieved without merging. Using

[5] The procedure you developed above for approximating derivatives cannot be used for your estimates based on the nested logit model. But because we have analytic expressions for market shares in the nested logit model, you could either differentiate these or use "finite difference" approximation of derivatives.

[6] If you get stuck working out the diversion ratios for nested logit – consult Conlon and Mortimer (2018).

[7] For your own benefit, you may want to see what happens without the approximation of the optimal instruments.

the `pyBLP` software, re-run the merger simulation with the 15% cost saving. Show the predicted post-merger price changes (again, for each product, averaged across markets). What is the predicted impact of the merger on consumer welfare,[8] assuming that the total measure of consumers M_t is the same in each market t?

15. Explain why this additional assumption (or data on the correct values of M_t) is needed here, whereas up to this point it was without loss to assume $M_t = 1$. What is the predicted impact of the merger on total welfare?

6 Coding Tips

- You can draw from a multivariate normal with variance Σ by drawing independent standard normal random variables and using the Cholesky decomposition of Σ (the latter obtained with `chol` in Matlab or `numpy.linalg.cholesky` in Python). You need to make sure you take the *lower triangular* portion. In particular, if $z = (z_1, \ldots, z_k)' \sim N(0, I_k)$ and $A = Chol(\Sigma)$, then Az is distributed $N(0, \Sigma)$.

- When you estimate the logit and nested logit models, you will have to choose which functions of the exogenous variables to use as instruments. One option would be to use all of them—the exogenous demand shifters (own and competing products) and the exogenous cost shifters. Alternatively, you might want to use something more like the BLP approximation of the optimal instruments. For example, for good j in market t, the instruments might be $x_{jt}, \mathrm{w}_{jt}, satellite_{jt}, wired_{jt}$, the quality index summed over competing goods $-j$ in market t, the quality index of the other good in the same nest as good j.[9] The `pyBLP` package gives some convenient built-in functions `pyblp.build_blp_instruments()` and `pyblp.build_differentiation_instruments()`.

  Of course, you do not need to feed an approximation of the optimal IV into `pyBLP` if you are going to have `pyBLP` compute a (better) approximation of the optimal instruments. Though the quality of the approximation depends on the initial set of instruments.

- To migrate your data from Matlab/R/Julia to Python, try exporting and importing a csv (i.e., comma separated) file. To export the data to a csv file, look into the `writematrix` or `writetable` functions in Matlab. To import the csv file, look into `pandas.read_csv` in Python.

- To display the average prices, use the following (where `changed_prices` is the output of `compute_prices` as in Post-Estimation Tutorial of pyBLP).

```
T, J= 600, 4
print(changed_prices.reshape((T, J)).mean(axis= 0))
```

[8] Note that because we have quasilinear preferences, consumer surplus is a valid measure of aggregate consumer welfare under the usual assumption of optimal redistribution.

[9] Note that on the supply side, there is an intercept in the marginal cost function, which will be collinear with the two dummies for satellite and wired.

- To display the average elasticities and diversion rations, use the following (where `elasticities`, for example, is the output of `compute_elasticities` in Post-Estimation Tutorial of pyBLP).

  ```
  T, J= 600, 4
  print(elasticities.reshape((T, J, J)).mean(axis= 0))
  ```

  (These resemble what one would write in Matlab, but there are subtle issues behind it, including row-major order (Python) vs column-major order (Matlab).)

- To apply 15% cost reduction by the merged firms, use the following.

  ```
  merger_costs= costs.copy()
  merger_costs[product_data.merger_ids== 1]= 0.85*merger_costs[product_data.merger_ids== 1]
  ```

  where `costs` and `merger_ids` are as in Post-Estimation Tutorial of pyBLP. (Using `merger_costs= costs` instead of using `copy` could lead to an unexpected behavior.)