Empirical Industrial Organization
Fall 2022 - Problem Set 1

Instructor: Cristián Sánchez

Due date: September 7, 2022

Estimating Preferences for Memory Chips

Download the dataset **choiceset_data.csv**. The data contain (fictitious) individual level purchase data from an online trading site for memory chips. On the site there are 48 possible different types of chip (a type is a size-speed-branded combination) and one seller for each type who can change his price over time. The data shows the characteristics of each type of chip when the customer visits, including a variable that indicates if the chip was not available (in which case you should not include this type in the choice set when you estimate).

The following is the description of the variables in the data set. The order of the variables in the description follows their order in the data set.

- VISITID: an ID number indicating the visit of a particular customer (e.g. Horace Andy visiting at 11:01pm). For each visit there are a number of rows, one for each of the product types available when the visit occured.
- SIZEMEM: the memory capacity of the chip (six different values, 1–6 representing e.g. 512MB to 16GB).
- SPEED: speed (four different values, 1–4).
- BRANDED: indicator for whether memory is branded or generic.
- PRICE: seller's list price.
- OUTOFSTOCK: indicator for whether the chip was not available for purchase.
- WHOLECOST: wholesale cost of the seller.

- CHOICE: indicator for whether the visitor bought the product (assume that she can only buy one unit). If no product is bought, then the consumer chose the outside good.

Note that you will need to perform additional data manipulation to do the computation. For instance, you will have to drop products that are out of stock at the moment of the visit, and you will have to incorporate the choice of the outside good.

Also, I want you to explicitly code your programs (e.g. likelihood function, gradient, hessian) when performing nonlinear estimation/optimization. Only if you are asked to perform linear estimation you are allowed to use already written estimation commands (e.g. regress, ivregress, ivreg2 in STATA). Always report standard errors along with your point estimates.

Aggregate Data

This section utilizes aggregate data. We are interested in the following preferences model:

$$U_{ij} = \beta_1 p_j + \beta_2' X_j + \xi_j + \varepsilon_{ij},$$

where the subscript ij refers to consumer i, product j. The variable p_j denotes the log of the price of the product. The vector X_j groups exogenous characteristics that include MEMSIZE (discrete bins/dummies), MEMSPEED (discrete bins/dummies), BRANDED (dummies). As usual, ξ_j captures product's unobserved (to the econometrician) characteristics that are relevant for the costumer's decision. Finally, ε_{ij} is an idiosyncratic Type I Extreme Value error term. The utility from buying the outside good is just ε_{i0}.

1. Aggregate the data at the product level. Construct market shares.
2. Using the inversion for the multinomial logit model with aggregate data seen in class, estimate the model by OLS. Do not forget to include a constant in your model. Report and comment your estimates.
3. Estimate the model by 2SLS, instrumenting the log of price with WHOLECOST. Report and comment your results. Compare your estimates with those obtained by OLS.

Individual Level Data

This section utilizes the data at the individual level. We are interested in the following preferences model:

$$U_{ij} = \beta_1 p_j + \beta_2' X_j + \varepsilon_{ij},$$

where we have abstracted from the existence of any unobserved product characteristic, ξ_j.

Note that, with individual level data, the constant in X_{jt} is 1 everywhere, except for the outside good, where it is 0.

We will estimate this model by maximum likelihood, using different optimization routines. We want to estimate the performance of the estimator.

1. Estimate the model using the Nelder-Mead (simplex) algorithm for optimization.
2. Estimate the model using any derivative-based algorithm for optimization (e.g. Newton). Use numerical derivatives to approximate both the gradient and the hessian.
3. Estimate the model using any derivative-based algorithm for optimization (e.g. Newton). Code the analytical gradient, and use it as part of your optimization routine. Use the BFGS approximation for the hessian.
4. Estimate the model using any derivative-based algorithm for optimization (e.g. Newton). Code the analytical gradient, as well as the analytical hessian, and use them as part of your optimization routine.
5. Report the estimates (along with standard errors), and the time each routine took to perform estimation. Comment.