ECON 899b - Problem Set 3

Due Monday, November 25th

Consider the following aggregate demand model with (quality) differentiation:

$$\begin{aligned} u_{ij,t} &= x_{j,t}\beta_x + \underbrace{\alpha_i}_{\alpha+\lambda_p y_i} p_{j,t} + \xi_{j,t} + \epsilon_{ij,t} && (1) \\ &= \delta_{j,t} + \mu_{ij,t} + \epsilon_{ij,t} \\ \rightarrow \sigma_j(\delta_t, p_t|\lambda_p) &\approx \frac{1}{R}\sum_r \frac{\exp(\delta_{j,t} + \mu_{rj,t})}{1 + \sum_{j'} \exp(\delta_{j',t} + \mu_{rj',t})} && (2) \end{aligned}$$

where $\mu_{ij,t} = \lambda_p y_i p_{j,t}$ and $u_{i0,t} = \epsilon_{i0,t}$, and y_i is a simulated vector of random coefficients (log income). The number of products in market t is J_t. Let $X_{jt} = (x_{jt}, p_{jt})$ and $\beta = (\beta_x, \alpha)$ denotes the combined vector of linear characteristics and parameters.

The derivative of the demand with respect to δ_{jt} (Jacobian matrix) is given by:

$$D\sigma_{j,k}(\delta_t, p_t|\lambda_p) = \begin{cases} \frac{1}{R}\sum_r \sigma_{r,j}(\delta_t, p_t|\lambda_p)(1 - \sigma_{r,j}(\delta_t, p_t|\lambda_p)) & \text{If } j = k \\ \frac{1}{R}\sum_r -\sigma_{r,j}(\delta_t, p_t|\lambda_p)\sigma_{r,k}(\delta_t, p_t|\lambda_p) & \text{If } j \neq k \end{cases}$$

where $\sigma_{r,j}(\delta_t, p_t|\lambda_p)$ is the choice probability of individual of type r. In matrix form, the Jacobian can be written as:

$$\Delta_t^{\sigma} = (1/R) \cdot I \cdot \left[\boldsymbol{\sigma}(1 - \boldsymbol{\sigma})^T\right] - (1/R) \cdot (1 - I) \cdot \left[\boldsymbol{\sigma}\boldsymbol{\sigma}^T\right]$$

where I is a $J_t \times J_t$ identity matrix, $\boldsymbol{\sigma}$ is a $J_t \times R$ matrix of choice probabilities, and $\cdot$ is an element-by-element product operator.

The residual function is defined by the inverse-demand function:

$$\rho_j(s_t, p_t|\lambda_p) = \sigma_j^{-1}(s_t, p_t|\lambda_p) - X_{j,t}\beta - \alpha p_{j,t}$$

where s_t is the observed market share vector. Using the implicit function theorem, we can define the derivative of the inverse-demand with respect to δ_t:

$$\Delta_t^{\delta} = -\left[\Delta_t^{\sigma}\right]^{-1}$$

The parameters are estimated using instrument vector Z to form moment conditions:

$$\begin{aligned} \min_{\lambda} \quad & \rho(s, p|\lambda_p)^T Z W Z^T \rho(s, p|\lambda_p) \\ \text{s.t.} \quad & \rho_j(s_t, p_t|\lambda_p) = \sigma_j^{-1}(s_t, p_t|\lambda_p) - X_{j,t}\beta^{iv} \\ & \beta^{iv} = \left((X'Z)W(Z'X)\right)^{-1}(X'Z)WZ'\boldsymbol{\delta}(\lambda_p) \end{aligned}$$

where $\boldsymbol{\delta}_{j,t} = \sigma_j^{-1}(s_t, p_t|\lambda_p)$.

You need to estimate the model using data from the US car industry covering the period between 1985 and 2015. To estimate the model you will need three data files:

```
Car_demand_characteristics_spec1.dta
Car_demand_iv_spec1.dta
Simulated_type_distribution.dta
```

The first file includes car characteristics: price, dollar-per-miles, HP/Weight, size (inches/1000), Turbo indicator, automatic transmission. The data also includes fixed effects for year, model class, number of cylinder, drive, and intercept. The second file includes the instruments: indicator variable if the car is assembled abroad (cost shifter), Euclidian distance (predicted price), and number competitors in four difference distance binds (predicted price dimension). The two files also include product and market variables: model_id and Year. The third file is $100x1$ vector of simulated income (random coefficient).

The Ox code for this problem is available on Canvas. You can use this code as an example to help write your own estimation routines.

1. Write a routine that invert the demand function for parameter value $\lambda_p = 0.6$. Invert the demand for the first year in the sample (1985) using two algorithm: contraction mapping, and a combination of contraction mapping and Newton. Plot the evolution of the norm between log predicted and observed shares across the iterations. Using a convergence threshold of $\varepsilon = 10^{-12}$, and set the threshold to start the Newton algorithm at $\varepsilon_1 = 1$.

2. Perform a grid search over the non-linear parameter: $\lambda_p \in [0, 1]$. Plot the GMM objective function as a function of the parameter λ_p. Use the 2SLS weighting matrix: $W = (Z^T Z)^{-1}$.

3. Using the minimum from the grid search, estimate the parameter λ_p using 2-step GMM. You can use the BFGS algorithm to minimize the function. The second-stage weighting matrix is given by: $W = \left[(Z \cdot \hat{\xi})(Z \cdot \hat{\xi})^T\right]^{-1}$ where $\hat{\xi} = \rho(s, p|\hat{\lambda})$.