BLP Demand Side Estimation

Zhang Donglei
CUHK Business School

March 28, 2017

1 Introduction

Using automobile data from 3 European countries (France, Germany, and Italy) over 1970-1999 (30 years), we estimated a random-coefficient logit model. In this study, each year-country combination is defined as a market. The market size is defined as the number of households in each market.

Six different optimization algorithms and five sets of starting values are employed in the estimation. This optimization design gives us 30 different optimization combinations, enabling us to derive deeper understanding of the BLP method from empirical perspective.

2 Data

2.1 Summary Statistics

The original dataset consists of prices, sales, and physical characteristics of all car models sold in three European countries (France, Germany and Italy) from 1970 to 1999. The total number of observations is 6,575. The data also includes variables to identify the model, the brand, the firm, the country of origin/production location, and the class (market segment). The dataset is augmented with macro-economic variables including population, exchange rates, GDP and consumer price indexes for the markets over the period. Table 1 provides summary statistics on the main variables, with the means also broken down by markets.

Table 1: Summary Statistics: Original Dataset

Mean	All	St.Dev.(all)	France	Germany	Italy
Sales (thousand units)	26.28	45.11	23.31	31.00	24.29
Population (million)	60.3	8.33	55.4	68.5	56.6
Horsepower (kW)	56.93	24.04	56.17	57.45	57.22
Horsepower per weight (hpwt, kW/kg)	0.056	0.013	0.06	0.06	0.06
Class (cla, 1-5 subcompact - luxury)	2.54	1.30	2.53	2.58	2.50
Price (relative to per capita income)	0.79	0.39	0.75	0.65	0.99
Domestic car (home, 0-1)	0.25	0.43	0.25	0.27	0.24
Average Fuel Efficiency (li, liter per km)	8.15	1.70	8.12	8.24	8.08
Maximum speed (km/h)	159.88	24.10	159.02	161.17	159.38
Time to acceleration (seconds)	15.32	5.32	15.39	15.09	15.48

In our model, we define market as year-country combination. There are 30 years and 3 countries, thus we have 90 markets. `x1` is a matrix contains variables enter the linear part of the estimation, consisting of a price variable (first column) , `hpwt`, `home`, `cla` and `li`. `x2` is a matrix contains all variables in `x1` except for `li`. These variables enter the non-linear part of the estimation. The matrix `v` contains random draws given for the unobserved characteristics for each market. For each market 200 *i.i.d.* $N(0,1)$ random draws are provided. They correspond to 50 individuals (ns=50), for each individual there is a different draw for each column of `x2`. Therefore, `v` is a 90×250 matrix for all countries.

2.2 Demographic Variables

The summary of demographic variables are shown in Table 2. The first two columns are the are the data in 1970, and the right two columns are the data in 1999. Demographic data is random draw of four demographic variables (household income, household income square, household size, and annual mileage) based on the variable mean and variance we collected in each market. For each market, there are 50 individuals. Therefore, `demogr` is a $6,575 \times 200$ matrix for all countries.

Table 2: Summary Statistics: Demographic Data

Country		Mean	St.D.	Mean	St.D.
France	Income	2884.5	2370	19481.6	2370
	Size	2.98	1.59	2.42	1.28
	Mileage	16	13.72	16	13.72
Germany	Income	3228.60	2370	21452.9	2370
	Size	2.76	1.59	2.20	1.28
	Mileage	13.5	13.72	13.5	13.72
Italy	Income	2981.15	2370	19621.8	2370
	Size	2.63	1.59	2.63	1.28
	Mileage	15	13.72	15	13.72

2.3 Instrumental Variables

In the case of automobiles, our identification strategy is similar to that in BLP(1995) and Knittel and Metaxoglou (2008). Our instruments consist of the five non-price automobile characteristics (constant, horsepower per weight, domestic car dummy, class, and average fuel efficiency), their sums across other automobiles produced by the same firm, as well as their sums across automobiles produced by the rival firms.

3 Estimation

3.1 Optimization Algorithms

No. 2, 3, 4, 5, 7, 8, and 9 optimization routines were chosen to estimate the parameters, that is Nelder-Mead Simplex, SolvOpt, Conjugate Gradient, Quasi-Newton 2, MADS, and GPS algorithms. The reason why we do not adopt No. 1, 6, and 10 optimization routines is that local minimum result will be obtain in No. 1 optimization routine, and with our data optimal value of the objective function can not be obtained in No. 6 and 10 optimization routines.

- **Nelder-Mead Simplex (routine 2):** `fminsearch.m` This algorithm uses a simplex of $n+1$ points for n-dimensional vectors x. The algorithm first makes a simplex around the initial guess x_0 by adding 5% of each component $x_0(i)$ to x_0, and using these n vectors as elements of the simplex in addition to x_0. (It uses 0.00025 as component i if $x_0(i) = 0$.) Then, the algorithm modifies the simplex repeatedly according to the following procedure.

- **SolvOpt (routine 3):** `SOLVOPT.m` SolvOpt (solver for local nonlinear optimization problems) is an implementation of Shor's r-algorithm by A. Kuntsevich and F. Kappel. In SolvOpt one can select to use either original subgradients or difference approximations of them (i.e. the user does not have to code difference approximations but to select one parameter to do this automatically). The constraints are taken into account by the method of exact penalization.

- **Conjugate Gradient (routine 4):** `conj_grad.m` The Conjugate Gradient method is an algorithm for finding the nearest local minimum of a function of n variables which presupposes that the gradient of the function can be computed. It uses conjugate directions instead of the local gradient for going downhill. If the vicinity of the minimum has the shape of a long, narrow valley, the minimum is reached in far fewer steps than would be the case using the method of steepest descent.

- **Quasi-Newton 2 (routine 5):** `ucminf.m` This is an algorithm for general-purpose unconstrained non-linear optimization. The algorithm is of quasi-Newton type with BFGS updating of the inverse Hessian and soft line search with a trust region type monitoring of the input to the line search algorithm.

- **GPS (routine 9):** The GPS algorithm uses fixed direction vectors. At each step, the algorithm searches a set of points, called a mesh, around the current point the point computed at the previous step of the algorithm. The mesh is formed by adding the current point to a scalar multiple of a set of vectors called a pattern. If the pattern search algorithm finds a point in the mesh that improves the objective function at the current point, the new point becomes the current poi nt at the next step of the algorithm.

- **MADS (routine 8):** The basic idea of MADS is the same as GPS, however, the MADS algorithm uses a random selection of vectors to dene the mesh.

3.2 Search Settings

With this program, data in different countries can be used by changing the number in the name of the data file to be loaded in line 20 (2 for France, 3 for Germany, and 4 for Italy). Different optimization routines (algorithms) can be applied by changing the number in line 36 from 1 to 10. In every optimization routine, we tried five different random starting values of θ_2 (`perturbs` in line 38).

Both termination tolerance on the function value (`TolFun`) and termination tolerance on X (`TolX`) are 1e-3. The maximum number of iterations (`MaxIter`) is 500,000 and the maximum number of function evaluations allowed within the solver (`MaxFunEvals`) is 4000. The starting value given to $theta2w$ is obtained by random draw.

4 Appendix: Code description

4.1 Scripts and Functions

The program is based on the code by Aviv Nevo. Knittel and Metaxoglou (2008) modified the code so that different starting values of θ_2 can be generated randomly and different algorithms can be used when estimating the GMM parameters. To use this program, just run `main.m`. After finishing the estimation, we can just run `optim_results_summary.m` first and then `Read optimization results.do` in `Optimization results` folder to obtain the summary of all estimation results.

The program consists of the following Matlab m-files and a Stata do-file:

- `main.m` - A script file that reads in the data and calls the other functions;

- `gmmobj.m` - This function computes the GMM objective function for Nelder-Mead Simplex, GA-JBES, MADS, GPS, GA-GADS, and Simulated Annealing algorithm;

- `gmmobj2.m` - This function computes the GMM objective function for Quasi-Newton 1 and SolvOpt algorithm;

- `gmmobj3.m` - This function computes the GMM objective function for Conjugate gradient and Quasi-Newton 2 algorithm;

- `gradobj.m` - This function computes the gradient of the GMM objective function;

- `meanval.m` - This function computes the mean utility level;

- `mufunc.m` - This function computes the non-linear part of the utility (μ_{ijt});
- `mktsh.m` - This function computes the market share for each product;
- `ind_sh.m` - This function computes the "individual" probabilities of choosing each brand;
- `jacob.m` - This function computes the Jacobian of the implicit function that defines the mean utility δ_{jt};
- `var_cov.m` - This function computes the variance-covariance matrix of the estimates;
- `printm.m` - This function prints the estimation results in different optimization routines;
- `optim_results_summary.m` - A script file that generates necessary `csv` file for optimization results summary by Stata;
- `Read optimization results.do` - This Stata do-file is in `Optimization results` folder and should be run after `optim_results_summary.m`. It can generate an xls-workbook concludes almost all estimation results such as the optimal values of objective function and θ.

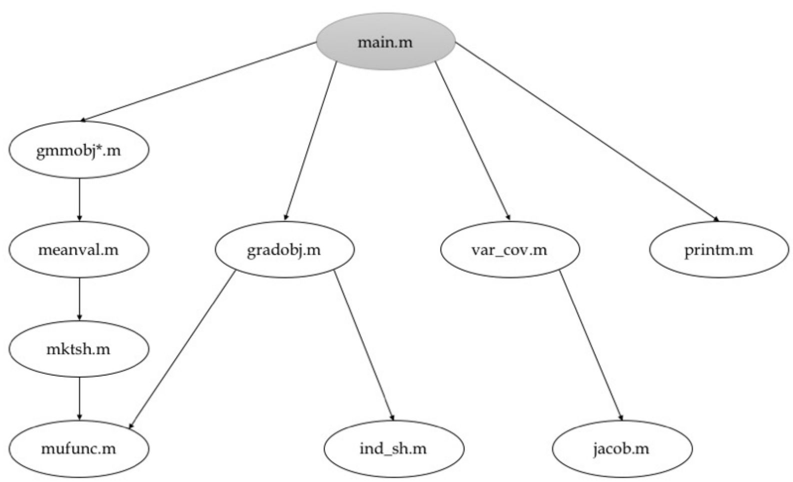

Figure 1: Program Framework

4.2 Some Comments

- We tried to include some interaction terms with demographic data. But when we loose the restrictions of the starting value of *theta2w* in line 162-166 (no longer set the right four columns zero forcedly), no reasonable objective function value can be obtained.
- We tried to include the brand dummies in $X1$. However, once the the brand dummies in $X1$ are included, no reasonable objective function value can be obtained.