Berry, Levinsohn & Pakes (1995 'BLP')

Sophie Bindslev and Anders Munk-Nielsen

2024 Fall

The purpose of this note is to provie a brief overview of the key equations of BLP, as needed to understand the implementation in the notebook from this folder. Not all details are covered: the reader is referred to Conlon & Gortmaker (2020) and to the PyBLP documentation on pyblp.readthedocs.io.

Notation	Description
i	Indexes individuals
j	Indexes products
t	Indexes markets
p_{jt}	The price of product j in market $t
$\mathbf{x}_{jt}$	$K \times 1$, of product characteristics
$\mathbf{z}_{it}$	$L \times 1$, of (demand) instruments (e.g., marginal cost shifters)
$\boldsymbol{\beta}_i$	$K \times 1$, of individual-specific coefficients
$\bar{\boldsymbol{\beta}}$	The mean of the distribution of $\boldsymbol{\beta}_i$
Σ	The covariance matrix of $\boldsymbol{\beta}_i$
δ_{jt}	The mean utility of product j in market $t
μ_{ijt}	The individual-specific deviation from the mean utility
ε_{ijt}	The *idiosyncratic* random utility shock
α	The price coefficient
ξ_{jt}	The unobserved demand shifter
θ	The vector of all parameters, $\theta = (\theta_1, \theta_2)$
θ_1	The linear parameters: $\theta_1 = (\bar{\boldsymbol{\beta}}, \alpha)$
θ_2	The non-linear parameters: $\theta_2 = \text{vec}(\Sigma)$

The Model

Individual level utility

Random Utility to individual i, from each option, j, in market, t:

$$u_{ijt} = \mathbf{x}'_{jt}\boldsymbol{\beta}_i + \alpha p_{jt} + \xi_{jt} + \varepsilon_{ijt}, \quad \varepsilon_{ijt} \sim \text{IID EVT1},$$

where EVT1 stands for Extreme Value Type I. As usual, the outside option is normalized to $u_{i0t} = \varepsilon_{i0t}$. Thus, the Conditional Choice Probabilities (CCP) become

$$P_{ijt} \equiv \Pr(i \text{ chooses } j \text{ in market } t) = \frac{\exp(\mathbf{x}'_{jt}\boldsymbol{\beta}_i + \alpha p_{jt} + \xi_{jt})}{1 + \sum_{k=1}^{J_t} \mathbf{x}'_{kt}\boldsymbol{\beta}_i + \alpha p_{kt} + \xi_{kt}}.$$

We assume that the rancom coefficients are distributed $\boldsymbol{\beta}_i \sim N(\bar{\boldsymbol{\beta}}, \Sigma)$, where $\bar{\boldsymbol{\beta}} \in \mathbb{R}^K$ is the mean of the distribution and Σ is the covariance matrix. If $\Sigma = 0_{K\times K}$, then we have the standard "vanilla" logit model. In this problem set, we will be working with *independent* random coefficients, i.e.

$$\Sigma = \text{diag}(\sigma_\beta^2).$$

This means that we can also write the coefficients as

$$\beta_{ik} = \bar{\beta}_{0k} + \sigma_{\beta k}\nu_{ik}, \quad \nu_{ik} \sim N(0,1).$$

With this in hand, it becomes convenient to decompose utility into a common *mean utility* part, δ_{jt}, and an individual-specific *deviation* from the mean, μ_{ijt}, that is mean-zero:

$$\begin{aligned} u_{ijt} &= \delta_{jt} + \mu_{ijt} + \varepsilon_{ijt} \\ \delta_{jt} &= \mathbf{x}'_{jt}\bar{\boldsymbol{\beta}} + \alpha p_{jt} + \xi_{jt} \\ \mu_{ijt} &= \mathbf{x}'_{jt}\Sigma\boldsymbol{\nu}_i = \sum_{k=1}^{K} \sigma_{\beta k}\nu_{ik}x_{jkt} \end{aligned}$$

Note: $\mathbb{E}[\mu_{ijt}] = 0$ because $\mathbb{E}[\nu_{ik}] = 0$. Also note that μ_{ijt} can still be written as $\mathbf{x}'_{jt}\Sigma\boldsymbol{\nu}_i$ when Σ is non-diagonal (i.e. when the coefficients are correlated), but writing it out as a sum with a single index utilizes that Σ is diagonal.

Now, we can write P_{ijt} in terms of the mean utilities, δ_{jt}, and the individual-specific part of the utilities, μ_{ijt}, for *all* products $j = 1, ..., J_t$ in market t:

$$\begin{aligned} &P_{ijt}(\boldsymbol{\delta}_t, \boldsymbol{\mu}_{it}) \\ &\qquad = \frac{\exp(\delta_{jt} + \mu_{ijt})}{1 + \sum_h^{J_t} \exp(\delta_{ht} + \mu_{iht})} \end{aligned}$$

Note in particular that P_{ijt} depends on the δ and μ terms for *all* products, not just for j, because they enter in the denominator.

Market Shares

Next, the challenge becomes that we only observe *market shares* and not individual choices. Hence, we will have to "average" (or "integrate") out the individual-specific part of the utilities, μ_{ijt}, to get the market shares.

Let $s_{jt}(\boldsymbol{\delta}_t, \theta_2)$ be the market share of product j in market t. It depends on the vector of mean utilities, $\boldsymbol{\delta}_t$, and the non-linear parameters, θ_2, which index the ("mixing") distribution of the idiosyncratic components μ_{ijt}. Technically, s_{jt} is the expected value of the choice probabilities with respect to the J_t vector $\boldsymbol{\mu}_{it}$, which is distributed according to $f(\cdot|\boldsymbol{\theta}_2)$. Mathematically,

$$\begin{aligned} s_{jt}(\boldsymbol{\delta}_t, \theta_2) &= \mathbb{E}_{\boldsymbol{\mu}_{it}}\left[P_{ijt}(\boldsymbol{\delta}_t, \boldsymbol{\mu}_{it})\right] \\ &= \int \frac{\exp(\delta_{jt} + \mu_j)}{1 + \sum_{k=1}^{J_t} \exp(\delta_{kt} + \mu_k)} f(\mu_1, \mu_2, ..., \mu_{J_t} | \boldsymbol{\theta}_2, \mathbf{x}_{jt}) d\boldsymbol{\mu}_{it}. \\ &= \int \frac{\exp(\delta_{jt} + \sum_k x_{jtk}\sigma_{\beta_k}\nu_k)}{1 + \sum_{\ell=1}^{J_t} \exp(\delta_{\ell t} + \sum_k x_{\ell tk}\sigma_{\beta k}\nu_k)} \phi(\nu_1) \cdot ... \cdot \phi(\nu_{J_t}) d\nu_1 ... d\nu_{J_t}, \end{aligned}$$

where ϕ is the standard normal density function. In the last equation, we have used the independence of the ν_k terms to write $f(\boldsymbol{\nu}) = \prod_k \phi(\nu_k)$.

Estimation

The coefficients are estimated by Generalised Method of Moments (GMM) using the moment conditions:

$$\mathbb{E}[\xi_{jt}\mathbf{z}_{it}(\boldsymbol{\beta})] = 0_{L\times 1}$$

where L is the number of moment conditions (i.e. $\mathbf{z}_{it}$ is Ł $\times$ 1 and ξ_{jt} is scalar).

Estimation is then conducted by replacing the expectation with a sample average and replacing the true error term, ξ_{jt}, with a residual, $\hat{\xi}_{jt}$. That is

$$\hat{\theta} = \arg\min_{\theta} g(\theta)' W g(\theta)$$

where the sample moments are

$$g(\theta) = \frac{1}{N} \sum_{t=1}^{T} \sum_{j=1}^{J_t} \hat{\xi}_{jt} \mathbf{z}_{jt}(\boldsymbol{\beta})$$

We will return to the choice of the weighting matrix, W, later. For now, we need a way of computing $\hat{\xi}_{jt}$, which is the "residual" from the demand inversion.

Nested Fixed Point: Concentrating out linear parameters

Before talking about demand inversion, let us consider a smart way of speeding up the minimization problem by splitting parameters into the linear, θ_1, and non-linear, θ_2, parameters. We can then estimate the non-linear parameters by GMM and for each trial value of the non-linear parameters, we can back out θ_1 by 2SLS.

For each trial value of non-linear parameters, θ_2, we do:

1. Invert demand: $\hat{\delta}_{jt} = D^{-1}(\mathcal{S}_t, \theta_2)$
2. 2SLS of $\hat{\delta}_{jt}$ on $(\mathbf{x}_{jt}, p_{jt})$ (using IVs $\mathbf{z}_{jt}$) to get $\boldsymbol{\theta}_1 = (\alpha, \boldsymbol{\beta}_x)$
3. $\hat{\xi}_{jt} = \hat{\delta}_{jt} - \mathbf{x}_{jt}\boldsymbol{\beta} - \alpha p_{jt}$ (the 2SLS residuals)
4. Compute $g(\theta_2)$

Demand inversion

Demand inversion is done using the Squarem algorithm, and it involves computing the integral over the random parameters (i.e. μ_{ijt}). It boils down to solving the fixed point problem:

$$\text{find } \boldsymbol{\delta}_t \text{ such that } s_{jt}(\boldsymbol{\delta}_t, \theta_2) = \mathcal{S}_{jt} \text{ for all } j = 1, ..., J_t)$$

where $\mathcal{S}_{jt}$ is the observed market share (data) of product j in market t.

Originally, BLP proposed iterating on the following contraction mapping:

$$\Gamma_t(\boldsymbol{\delta}_t) = \boldsymbol{\delta}_t + \log(\mathcal{S}_t) - \log(s_t(\boldsymbol{\delta}_t, \theta_2))$$

Clearly, if $\Gamma_t(\boldsymbol{\delta}_t) = \boldsymbol{\delta}_t$ we have solved the original problem. In other words, we are looking for a fixed point of $\Gamma_t(\cdot)$.

BLP showed that Γ is a contraction mapping: this means that the iterative scheme,

$$\boldsymbol{\delta}^r = \Gamma_t(\boldsymbol{\delta}^{r-1})$$

converges to the fixed point. However, the rate of convergence can be slow.

The Squarem Algorithm provides a fast alternative. The details are in `BLP.py` in the function `fp_squarem()`.

Weighting matrix

The choice of the $L \times L$ weighting matrix W does not affect *consistency*. That is, if the sample size goes to infinity, the GMM estimator will converge to the true parameter values. However, the choice of W does affect the *efficiency* of the estimator, i.e. the precision in finite samples.

The efficient weighting matrix is the inverse of the variance-covariance matrix of the moment conditions. This is unknown, but can be consistently estimated by the sample analogue of the variance-covariance matrix of the moment conditions, $\mathrm{Var}(\xi \mathbf{Z})$. The problem is that these depend on parameters, which are unknown. In practice, a two-step procedure is used:

1. Estimate the parameters by GMM using some initial weighting matrix, $W_0 = \widehat{\mathrm{Var}}(\mathbf{Z})$, its diagonal, or simply just the $L \times L$ identity matrix. This will still capture the first-order variation in the moment conditions, for example putting a lower weight on the instruments with a lot of variance.[1]
2. Given the residuals, we can comptue an empirical estimate of of the variance matrix of the moment conditions. The formula is somwhat messy, it is in the function `getCovariance()`.

[1]Of course, the variance of an instrument is not the only thing that matters. For one, it does not tell us whether the instrument is correlated with the endogeneous variable, p_{jt}. But we need a starting point.