‘MATH+ECON+CODE’ MASTERCLASS ON MATCHING MODELS, OPTIMAL TRANSPORT AND APPLICATIONS

Alfred Galichon (New York University)

Spring 2018
Day 4, January 18 2018: “Multinomial choice”
Block 11. Demand models, old and new

- Beyond GEV: the pure characteristics models, the random coefficient logit model, the probit model
- Simulation methods: AR, GHK, and SARS
- The inversion theorem

References for block 11

- [OTME], Ch. 9.2
- McFadden (1981). "Econometric Models of Probabilistic Choice," in C.F. Manski and D. McFadden (eds.), *Structural analysis of discrete data with econometric applications*, MIT Press.
- Berry, Levinsohn, and Pakes (1995). "Automobile Prices in Market Equilibrium," *Econometrica*.
- Berry and Pakes (2007). The pure characteristics demand model". *International Economic Review*
- Train. (2009). *Discrete Choice Methods with Simulation*. 2nd Edition. Cambridge University Press.

Section 1

THEORY

- The GEV models are convenient analytically, but not very flexible.
  - The logit model imposes zero correlation across alternatives
  - The nested logit allows for nonzero correlation, but in a very rigid way (needs to define nests).
- A good example is the probit model, where ε is a Gaussian vector. For this model, there is no close-form solution neither for G nor for G^*.
- More recently, a number of modern models don't have closed-form either. These models require simulation methods in order to approximate them by discrete models.

- The pure characteristics model (Berry and Pakes, 2007) can be motivated as follows. Assume y stands for the number of bedrooms. The logit model would assume that the random utility associated with a 2-BR is uncorrelated with a 3-BR, which is not realistic.
- Let ξ_y is the typical size of a bedroom of size y, one may introduce ϵ as the valuation of size; in which case the utility shock associated with y should be $\varepsilon_y = \epsilon \xi_y$. More generally, the characteristics ξ_y is a d-dimensional (deterministic) vector, and $\epsilon \sim \mathbf{P}_\epsilon$ is a (random) vector of the same size standing for the valuations of the respective dimensions, so that
$$\varepsilon_y = \epsilon^\intercal \xi_y.$$
- For example, if each alternative y stands for a model of car, the first component of ξ_y may be the price of car y; the other components may be other characteristics such as number of seats, fuel efficiency, size, etc. In that case, for a given dimension $y \in \mathcal{Y}_0$, ϵ_y is the (random) valuation of this dimension by the consumer with taste vector ϵ.

- Assume without loss of generality that $\varepsilon_y = 0$, that is $\xi_0 = 0$ as we can always reduce the setting to this case by replacing ξ_y by $\xi_y - \xi_0$.
- Letting Z be the $|\mathcal{Y}| \times d$ matrix of (y, k)-term ξ_y^k, this rewrites as

$$\varepsilon = Z\epsilon.$$

- Hence, we have

$$G(U) = \mathbb{E}\left[\max\left\{U + Z\epsilon, 0\right\}\right].$$

and

$$\sigma_y(U) = \Pr\left(U_y - U_z \geq (Z\epsilon)_y - (Z\epsilon)_z \ \ \forall z \in \mathcal{Y}_0 \backslash \{y\}\right).$$

- When $d = 1$ (scalar characteristics), one has $\sigma_y(U) = \Pr(U_y - U_z \geq (\xi_y - \xi_z)\epsilon \; \forall z \in \mathcal{Y}_0 \backslash \{y\})$, and thus

$$\sigma_y(U) = \Pr\left(\max_{z:\xi_y > \xi_z}\left\{\frac{U_y - U_z}{\xi_y - \xi_z}\right\} \leq \epsilon \leq \min_{z:\xi_y < \xi_z}\left\{\frac{U_y - U_z}{\xi_y - \xi_z}\right\}\right)$$

  with the understanding that $\max_{z \in \emptyset} f_z = -\infty$ and $\min_{z \in \emptyset} f_z = +\infty$.
- Therefore, letting $\mathbf{F}_\epsilon$ be the c.d.f. associated with the distribution of ϵ, one has a closed-form expression for σ_y:

$$\sigma_y(U) = \mathbf{F}_\epsilon\left(\left[\max_{z:\xi_y > \xi_z}\left\{\frac{U_y - U_z}{\xi_y - \xi_z}\right\}, \min_{z:\xi_y < \xi_z}\left\{\frac{U_y - U_z}{\xi_y - \xi_z}\right\}\right]\right)$$

- When $\mathbf{P}_\epsilon$ is the $\mathcal{N}(0, S)$ distribution, then the pure characteristics model is called a Probit model; in this case,

$$\varepsilon \sim \mathcal{N}(0, \Sigma) \text{ where } \Sigma = ZSZ^\intercal.$$

- Note the distribution ε will not have full support unless $d \geq |\mathcal{Y}|$ and Z is of full rank.
- Computing σ in the Probit model thus implies computing the mass assigned by the Gaussian distribution to rectangles of the type

$$[l_y, u_y].$$

When Σ is diagonal (random utility terms are i.i.d. across alternatives), this is numerically easy. However, this is computationally difficult in general (more on this later).

- The random coefficient logit model (Berry, Levinsohn and Pakes, 1995) may be viewed as an interpolant between the random characteristics model and the logit model. In this case,

$$\varepsilon = (1-\lambda)\, Z\epsilon + \lambda\eta$$

where $\epsilon \sim \mathbf{P}_\epsilon$, η is an EV1 distribution independent from the previous term, and λ is a interpolation parameter ($\lambda = 1$ is the logit model, and $\lambda = 0$ is the pure characteristics model).

- In this case, one may compute the Emax operator as

$$\begin{aligned} G(U) &= \mathbb{E}\left[\max_{y\in\mathcal{Y}_0}\left\{U_y + (1-\lambda)\,(Z\epsilon)_y + \lambda\eta_y\right\}\right] \\ &= \mathbb{E}\left[\mathbb{E}\left[\max_{y\in\mathcal{Y}_0}\left\{U_y + (1-\lambda)\,(Z\epsilon)_y + \lambda\eta_y\right\}|\epsilon\right]\right] \\ &= \mathbb{E}\left[\lambda\log\sum_{y\in\mathcal{Y}_0}\exp\left(\frac{U_y + (1-\lambda)\,(Z\epsilon)_y}{\lambda}\right)\right] \end{aligned}$$

- Recall

$$G(U) = \mathbb{E}\left[\lambda \log \sum_{y \in \mathcal{Y}_0} \exp\left(\frac{U_y + (1-\lambda)(Z\epsilon)_y}{\lambda}\right)\right].$$

- The demand map in the random coefficients logit model is obtained by derivation of the expression of the Emax, i.e.

$$\sigma_y(U) = \mathbb{E}\left[\frac{\exp\left(\frac{U_y+(1-\lambda)(Z\epsilon)_y}{\lambda}\right)}{\sum_{y' \in \mathcal{Y}_0} \exp\left(\frac{U_{y'}+(1-\lambda)(Z\epsilon)_{y'}}{\lambda}\right)}\right].$$

- In a number of cases, one cannot compute the choice probabilities $\sigma(U)$ using a closed-form expression. In this case, we need to resort to simulation to compute G, G^*, σ and σ^{-1}.
- The idea is that:
  - one is able to compute G and G^* for discrete distributions (more on this later)
  - the sampled versions of G, G^*, σ and σ^{-1} converge to the populations objects when the sample size is large.

- One simulates N points $\varepsilon^i \sim P$. The Emax operator associated with the empirical sample distribution P_N is

$$G_N = N^{-1} \sum_{i=1}^{N} \max_{y \in \mathcal{Y}} \left\{ U_y + \varepsilon_y^i \right\}$$

and the demand map is given by

$$\sigma_{N,y}(U) = N^{-1} \sum_{i=1}^{N} 1 \left\{ U_y + \varepsilon_y^i \geq U_z + \varepsilon_z^i \ \forall z \in \mathcal{Y}_0 \right\}$$

- In the literature, σ_N is called the *accept-reject simulator.*

- GHK simulator (Geweke, Hajivassiliou and Keane) improves on the AR simulator.
- Without loss of generality, we shall focus on the market share of $y = 0$, and label the elements of $\mathcal{Y}$ as $\mathcal{Y} = \{1, ..., M\}$ where $M = |\mathcal{Y}|$. Then if $\mathbf{F}_\eta$ is the c.d.f. of the random vector η valued in $\mathbb{R}^{\mathcal{Y}}$ defined by $\eta_y := \varepsilon_y - \varepsilon_0$ for $z \in \mathcal{Y}$, then

$$\mu_0 = 1 - 1'_{\mathcal{Y}} \nabla G\left(U\right) = \mathbf{F}_\eta\left(z\right),$$

where $z_y = -U_y$, for all $y \in \mathcal{Y}$.

- Note that $\mathbf{F}_\eta\left(z\right) = \Pr\left(\eta_1 \leq z_1, ..., \eta_M \leq z_M\right)$ can be expressed as

$$\mu_0 = \prod_{j=1}^{M} \Pr\left(\eta_i \leq z_i | \eta_1 \leq z_1, ...\eta_{j-1} \leq z_{j-1}\right) \tag{1}$$

with the understanding that the first term in this product (associated with $j = 1$) is simply the unconditional probability distribution $\Pr\left(\eta_1 \leq z_1\right)$.

- The fundamental assumption behind the GHK method is that the Rosenblatt quantile associated with the distribution of η is known.
- Recall from the exercise discussed in the previous lecture that the Rosenblatt quantile is the map T such that $T\#\mu = P$, and such that the Jacobian DT of T is lower triangular with nonnegative diagonal. That is,

$$\begin{cases} \eta_1 = T_1(U_1) \\ \eta_2 = T_2(U_1, U_2) \\ ... \\ \eta_M = T_M(U_1, U_2, ..., U_M) \end{cases},$$

where $U \sim U([0,1]^d)$, and $T_i(u)$ depends only on u_1,...,u_i and is a nondecreasing function of u_i. In order to evaluate quantity $\mathbf{F}_\eta(z)$ in (1), one needs to evaluate $\pi_j = \Pr(\eta_j \leq z_j | \eta_1 \leq z_1, ...\eta_{j-1} \leq z_{j-1})$, that is

$$\pi_j = \mathbb{E}\left[1\left\{T_j(U_1, U_2, ..., U_j) \leq z_i\right\} | T_1(U_1) \leq z_1, ..., T_{j-1}(U_1, U_2, ..., U_{j-1})\right. \tag{2}$$

- Denote $T_i^{-1}\left(z;u_1,...,u_{i-1}\right)$ the inverse of $u_i \mapsto T_i^{-1}\left(u_1,...,u_{i-1},u_i\right)$ for fixed values of u_1,...,u_{i-1}.
- Then, observe that if $\tilde{U} \sim U([0,1]^d)$, and if

$$\left\{ \begin{array}{l} \hat{U}_1 = \tilde{U}_1 T_1^{-1}\left(z_1\right) \\ \hat{U}_2 = \tilde{U}_2 T_2^{-1}\left(z_2;\hat{U}_1\right) \\ ... \\ \hat{U}_M = \tilde{U}_M T_M^{-1}\left(z_M;\hat{U}_1,...,\hat{U}_{M-1}\right) \end{array} \right. . \tag{3}$$

- Then the conditional expectation

$$\pi_j = \mathbb{E}\left[1\left\{T_j\left(U_1,U_2,...,U_j\right) \leq z_i\right\} | T_1\left(U_1\right) \leq z_1, T_2 \leq etc.\right]$$

coincides with the unconditional expectation

$$\pi_j = \mathbb{E}\left[T_j^{-1}\left(z_j;\hat{U}_1,\hat{U}_2,...,\hat{U}_{j-1}\right)\right]. \tag{4}$$

- The GHK simulator can be interpreted as an importance sampling simulation procedure.
- Indeed, expression

$$\pi_j = \mathbb{E}\left[1\left\{T_j\left(U_1, U_2, ..., U_j\right) \leq z_i\right\} | T_1\left(U_1\right) \leq z_1, T_2 \leq etc.\right]$$

  is a conditional expectation; one may compute it by accept-reject but this is computationally suboptimal, as it leads us to discard a fraction of draws – which can be a significant fraction.
- In contrast, in expression

$$\pi_j = \mathbb{E}\left[T_j^{-1}\left(z_j; \hat{U}_1, \hat{U}_2, ..., \hat{U}_{j-1}\right)\right].$$

  is an unconditional expectation; one shall compute it by drawing K i.i.d. draws of $\tilde{U} \sim U([0,1]^d)$, computing the $\hat{U}$'s, and averaging over all the values of $T_j^{-1}\left(\hat{U}_1, \hat{U}_2, ..., \hat{U}_{j-1}, z_i\right)$ simulated that way. In the second method, we have not discarded any draws, which is more efficient.

Algorithm.
For $k = 1, ...K$: Draw $\tilde{U}^k \sim \mathcal{U}([0,1]^d)$. Compute $(\hat{U}_1^k, ..., \hat{U}_M^k)$ from $(\tilde{U}_1^k, ..., \tilde{U}_M^k)$ using transformation (3). Compute $\pi_j^K = K^{-1} \sum_{k=1}^K T_j^{-1}\left(z_j; \hat{U}_1^k, \hat{U}_2^k, ..., \hat{U}_{j-1}^k\right)$ for $j = 1, ..., M$. Return the GHK simulator

$$\mu_0 = \prod_{j=1}^{M} \pi_j^K.$$

Remark: The practical difficulty with the implementation of this algorithm is the knowledge of the Rosenblatt quantile in closed form. A leading example where this object is readily available is given by the case when $\boldsymbol{P}$ is Gaussian, which is called the probit model.

- The Probit model is characterized by $\boldsymbol{P} = \mathcal{N}(0, \Sigma)$, with Σ is a $M \times M$ symmetric semidefinite positive matrix, hence $cov\left(\varepsilon_y, \varepsilon_{y'}\right) = \Sigma_{yy'}$. In this case, the Rosenblatt quantile is known. Let $\Sigma = LL^\intercal$ be the Choleski decomposition of Σ, where L is lower triangular with a positive diagonal. Then the Rosenblatt quantile T is such that

$$\begin{cases} T_1(u) = L_{11}\Phi^{-1}(u_1) \\ T_2(u) = L_{21}\Phi^{-1}(u_1) + L_{22}\Phi^{-1}(u_2) \\ ... \\ T_M(u) = L_{M1}\Phi^{-1}(u_1) + ... + L_{MM}\Phi^{-1}(u_M) \end{cases}$$

where Φ is the c.d.f. of the standard normal (univariate) distribution. In this case, $\hat{U}$ is obtained from $U \sim \mathcal{U}\left([0,1]^M\right)$ by

$$\begin{cases} \hat{U}_1 = \tilde{U}_1 \Phi\left(\frac{z_1}{L_{11}}\right) \\ \hat{U}_2 = \tilde{U}_2 \Phi\left(\frac{z_2 - L_{21}\Phi^{-1}\left(\hat{U}_1\right)}{L_{22}}\right) \\ ... \\ \hat{U}_M = \tilde{U}_M \Phi\left(\frac{z_M - L_{M1}\Phi^{-1}\left(\hat{U}_1\right) ... - L_{M(M-1)}\Phi^{-1}\left(\hat{U}_{M-1}\right)}{L_{MM}}\right) \end{cases}.$$

- McFadden's smoothed accept-reject simulator (SARS) consists in sampling $\varepsilon \sim P$: $\varepsilon^1, ..., \varepsilon^N$, and replacing the max by the smooth-max

$$\sigma_{N,T,y}(U) = \sum_{i=1}^{N} \frac{1}{N} \frac{\exp\left((U_y + \varepsilon_y^i)/T\right)}{\sum_z \exp\left((U_z + \varepsilon_z^i)/T\right)}$$

- One seeks U so that the induced choice probabilities are s, that is

$$s_y = \sum_{i=1}^{N} \frac{1}{N} \frac{\exp\left((U_y + \varepsilon_y^i)/T\right)}{\sum_z \exp\left((U_z + \varepsilon_z^i)/T\right)}.$$

- The associated Emax operator is

$$G_{N,T}(U) = \mathbb{E}_{\mathbf{P}_N}\left[G_{\text{logit}}\left(U + \varepsilon^i\right)\right]$$

so the underlying random utility structure is a random coefficient logit.

THEOREM

Consider a solution $(u(\varepsilon), v_y)$ to the dual Monge-Kantorovich problem with cost $\Phi(\varepsilon, y) = \varepsilon_y$, that is:

$$\min_{u,v} \int u(\varepsilon)\, d\mathbf{P}(\varepsilon) + \sum_{y \in \mathcal{Y}_0} v_y s_y \tag{5}$$

$$s.t.\ u(\varepsilon) + v_y \geq \Phi(\varepsilon, y)$$

Then:
(i) $U = \sigma^{-1}(s)$ is given by $U_y = v_0 - v_y$.
(ii) The value of Problem (5) is $-G^*(s)$.

PROOF.

$\sigma^{-1}(s) = \arg\max_{U:U_0=0} \left\{ \sum_{y\in\mathcal{Y}} s_y U_y - G(U) \right\}$, thus, letting $v = -U$, v is the solution ot

$$\min_{v:v_0=0} \left\{ \sum_{y\in\mathcal{Y}_0} s_y v_y + G(-v) \right\}$$

which is exactly problem (5). $\square$

- It follows from the inversion theorem that the problem of demand inversion in the pure characteristics model is a semi-discrete transport problem.
- Indeed, the correspondence is:
  - an alternative y is a fountain
  - the characteristics of an alternative is a fountain location
  - the systematic utility associated with alternative y is minus the price of fountain y
  - the market share of altenative y coindides with the capacity of fountain y
  - the random vector ϵ is the location of an inhabitant

- Let $u_i = T \log \sum_z \exp\left((U_z + \varepsilon_z^i) / T \right)$. One has

$$\left\{ \begin{array}{l} s_y = \sum_{i=1}^N \frac{1}{N} \exp\left((U_y - u_i + \varepsilon_y^i) / T \right) \\ \frac{1}{N} = \sum_y \frac{1}{N} \exp\left((U_y - u_i + \varepsilon_y^i) / T \right) \end{array} \right. .$$

- As a result, (u_i, U_y) are the solution of the regularized OT problem

$$\min_{u,U} \sum_{i=1}^N \frac{1}{N} u_i - \sum s_y U_y + \sum_{i,y} \frac{1}{N} \exp\left((U_y - u_i + \varepsilon_y^i) / T \right).$$

- Consider the IPFP algorithm for solving the latter problem:

$$\begin{cases} \exp\left(u_i^{k+1}/T\right) = \sum_z \exp\left(\left(U_z^k + \varepsilon_z^i\right)/T\right) \\ \exp U_y^{k+1}/T = \frac{N s_y}{\sum_{i=1}^N \exp\left(\left(-u_i^{k+1} + \varepsilon_y^i\right)/T\right)} \end{cases}$$

- This rewrites as

$$\exp U_y^{k+1}/T = \frac{N s_y}{\sum_{i=1}^N \frac{\exp\left(\varepsilon_y^i/T\right)}{\sum_z \exp\left(\left(U_z^k+\varepsilon_z^i\right)/T\right)}}, \text{ i.e.}$$

$$U_y^{k+1} = T \log s_y - T \log \sum_{i=1}^N \frac{1}{N} \frac{\exp\left(\varepsilon_y^i/T\right)}{\sum_z \exp\left(\left(U_z^k + \varepsilon_z^i\right)/T\right)}$$

which is exactly the contraction mapping algorithm of Berry, Levinsohn and Pakes (1995, appendix 1).

Section 2

Coding

- We shall code the AR simulator for the probit model and then invert it using the inversion theorem.
- Take a vector of systematic utilities:
  ```
  U_y = c(1.6, 3.2, 1.1,0)
  ```
- Simulate the market shares using the AR simulator:
  ```
  epsilon_iy = matrix(rnorm(nbDraws*nbY),ncol=nbY) %*%
  SqrtCovar
  u_iy = t(t(epsilon_iy)+U_y)
  ui = apply(X = u_iy, MARGIN = 1, FUN = max)
  s_y = apply(X = u_iy - ui, MARGIN = 2,FUN = function(v)
  (length(which(v==0)))) / nbDraws
  ```

- To invert the market share, simply run the optimal assignment problem:

```
A1 =
kronecker(matrix(1,1,nbY),sparseMatrix(1:nbDraws,1:nbDraws))
A2 =
kronecker(sparseMatrix(1:nbY,1:nbY),matrix(1,1,nbDraws))
A = rbind2(A1,A2)
result = gurobi (
list(A=A,obj=c(epsilon_iy),modelsense="max",
rhs=c(rep(1/nbDraws,nbDraws),s_y) ,sense="="),
params=list(OutputFlag=0) )
Uhat_y = - result$pi[(1+nbDraws):(nbY+nbDraws)] +
result$pi[(nbY+nbDraws)]
```