Improving the Numerical Performance of BLP Static and Dynamic Discrete Choice Random Coefficients Demand Estimation

Jean-Pierre Dubé[1]
Jeremy T. Fox[2]
Che-Lin Su[3]

[1]University of Chicago, Graduate School of Business
[2]University of Chicago, Dept. of Economics, and NBER
[3]University of Chicago, Graduate School of Business

FTC Conference
November 2008

BLP (1995) Demand Estimation

- Berry, Levinsohn and Pakes (1995) or “BLP” consists of an economic model and a GMM estimator
- Demand estimation with a large number of differentiated products
  - Product characteristics approach
  - Requires only aggregate market share data
  - Flexible substitution patterns / price elasticities
  - Controls for price endogeneity
- Computational algorithm to construct moment conditions from nonlinear model
- Useful for measuring market power, welfare, optimal pricing, etc.
- Used extensively in industrial organization and marketing
  - Nevo (2001), Petrin (2002), Sudhir (2002), ...

Computational concerns of BLP users and non-users

- Method, if it delivers, is clearly very useful
  - Not tons of good alternatives
  - Useful in antitrust, consulting, in addition to academic research
- Takes time to learn how to correctly code and use
- Typical applied user: no formal training in implementation?
  - BLP (1995) somewhat dense
  - Nevo (2000) has some advice
- Concern: reliability of empirical results
  - No point in using fancy estimator if you are going to report wrong estimates
  - Knittel & Metaxoglou (2008) alarmist message
  - New research on dynamic demand, up to four inner loops
    - Gowrisankaran & Rysman (2008), Lee (2008), Schiraldi (2008)
- Our broad goal: document some (computational) concerns and offer some solutions

BLP's estimation algorithm

- Nested Fixed Point (NFP) approach
  - Nest fixed point calculation (*inner loop*) into parameter search (*outer loop*)
- Propose contraction mapping to calculate fixed point
- Our concerns
  - Trade off inner loop numerical error versus speed
  - Error in inner loop propagates into outer loop
  - Wrong parameter estimates
- Concern regards NFP algorithm, not actual statistical properties of BLP
- Our solution is MPEC
  - Mathematical program with equilibrium constraints
  - MPEC & NFP are statistically the same estimator (Berry, Linton & Pakes 2004)
  - See Su & Judd (2008) for non-demand applications

Our contributions

1. Analyze numerical properties of the NFP algorithm
2. Poor implementation can lead to wrong parameter estimates
3. MPEC: alternative computational method
   - Impossible to have same numerical errors as NFP
   - Can execute faster than NFP
   - Applies to models where contraction mapping does not exist
     - Richer static models, Gandhi (2008)
     - Many forward-looking, dynamic demand models
     - Even models with multiple demand shocks to satisfy market shares?
4. Issues with NFP more severe in dynamic demand applications
   - Multiple nested loops
   - Bellman iterations more computationally expensive
   - MPEC's advantage may be even greater in these cases

Discrete choice demand model

$$u_{i,j,t} = {}^{0}_{i} + x^{0}_{j,t} \; {}^{x}_{i} \quad {}^{p}_{i} p_{j,t} + {}_{j,t} + "_{i,j,t}$$

- Consumer i, choice $j \; 2 \; J$, market $t \; 2 \; T$
- Product characteristics $x_{j,t}$, $p_{j,t}$, ${}_{j,t}$
    - ${}_{j;t}$ not in data
- ${}^{0}_{i}$, ${}^{x}_{i}$, ${}^{p}_{i}$ random coe□cients
    - Distribution $F \; (\; ; \;)$
    - BLP's statistical goal: estimate in parametric distribution
- ${}_{i,j,t}$ extreme value shock (logit)
- i picks j if $u_{i,j,t} \quad u_{i,k,t} 8 k \; 2 \; J; k \; 6\!= j$

Inner loop of NFP approach

- Compute numerically

$$(\ ) = s^{\ 1}(S; \ )$$

- BLP propose a contraction-mapping
  - For each guess iterate on

$$_t^{h+1}$$

Contraction Mapping Theorem

Some details skipped

- Assume that T is a contraction mapping:)

$$T(\quad)\quad T(\tilde{\;})\quad L(\quad)\ \tilde{)}$$

Lipschitz constant for BLP contraction mapping

- can show it's related to Jacobian of iteration operator

$$L = \max_{\xi \in D} \| I - \nabla_{\xi} (\log s(\xi;\theta)) \|,$$

where $\frac{\partial(\log s_{jt}(\xi;\theta))}{\partial \xi_{lt}}$ is, for $j = l$ and $j \neq l$ respectively

$$\frac{\sum_{r=1}^{ns} \left[\left(\frac{\exp\left(x_{jt}^{0}\beta^{r} - \alpha^{r} p_{jt} + \xi_{jt}\right)}{1 + \sum_{k=1}^{J} \exp\left(x_{kt}^{0}\beta^{r} - \alpha^{r} p_{kt} + \xi_{kt}\right)} \right) - \left(\frac{\exp\left(x_{jt}^{0}\beta^{r} - \alpha^{r} p_{jt} + \xi_{jt}\right)}{1 + \sum_{k=1}^{J} \exp\left(x_{kt}^{0}\beta^{r} - \alpha^{r} p_{kt} + \xi_{kt}\right)} \right)^{2} \right]}{\sum_{r=1}^{ns} \frac{\exp\left(x_{jt}^{0}\beta^{r} - \alpha^{r} p_{jt} + \xi_{jt}\right)}{1 + \sum_{k=1}^{J} \exp\left(x_{kt}^{0}\beta^{r} - \alpha^{r} p_{kt} + \xi_{kt}\right)}}$$

$$- \frac{\sum_{r=1}^{ns} \left[\left(\frac{\exp\left(x_{jt}^{0}\beta^{r} - \alpha^{r} p_{jt} + \xi_{jt}\right)}{1 + \sum_{k=1}^{J} \exp\left(x_{kt}^{0}\beta^{r} - \alpha^{r} p_{kt} + \xi_{kt}\right)} \right) \left(\frac{\exp\left(x_{lt}^{0}\beta^{r} - \alpha^{r} p_{lt} + \xi_{lt}\right)}{1 + \sum_{k=1}^{J} \exp\left(x_{kt}^{0}\beta^{r} - \alpha^{r} p_{kt} + \xi_{kt}\right)} \right) \right]}{\sum_{r=1}^{ns} \frac{\exp\left(x_{jt}^{0}\beta^{r} - \alpha^{r} p_{jt} + \xi_{jt}\right)}{1 + \sum_{k=1}^{J} \exp\left(x_{kt}^{0}\beta^{r} - \alpha^{r} p_{kt} + \xi_{kt}\right)}}$$

- Most scholars use smooth optimizers, which use gradient information
- Gradient often approximated by numerical derivatives

$$r_d Q(\ (\ ;\ _{\text{in}})) = \frac{Q(\ (\ + de_k;\ _{\text{in}})) \quad Q(\ (\quad de_k;\ _{\text{in}}))}{2d} \quad {}_{k=1}^{j\theta j}$$

- Gradient error bounded

$$k\, r_d Q(\ (\ ;\ _{\text{in}})) \quad r\, Q(\ (\ ;0))k_1 \quad O\ \ d^2 + \frac{1}{d} O \quad \frac{L(\ )}{1 \quad L(\ )}\ _{\text{in}}$$

- Search algorithm could go in wrong direction because of numerical error!

Simulated data setup

-

Simulation draws

- Goal is not to discuss error from numerical integration
- Use same 100 draws in numerical integrals in data creation and estimation
- No numerical error from integration
- In practice, multiply all computing times by 100
  - 10,000 draws
- Not clear fewer draws favors either NFP, MPEC

Software details

- MATLAB, highly vectorized code
  - Parallelizes well
- Optimization software KNITRO
  -

Nevo's cereal data: Loose versus tight tolerances for NFP

With closed-form derivatives

	NFP Loose Inner	NFP Loose Both	NFP Tight
Fraction Convergence	0.0	0.81	1.00
Frac. $<$ 1% $>$ "Global" Min.	0.0	0.0	1.00
Mean Own Price Elasticity	-3.75	-3.69	-7.43
Std. Dev. Own Price Elasticity	0.03	0.08	~0
Lowest Objective	15.3816	15.4107	4.5615
Elasticity for Lowest Obj.	-3.77	-3.77	-7.43

- Nevo (2000) cereal data (pseudo-real) – prices, quantities, characteristics across multiple markets
- 25 starting values
- NFP loose inner loop: $_{\text{in}} = 10^{4}$, $_{\text{out}} = 10^{6}$
- NFP loose both: $_{\text{in}} = 10^{4}$, $_{\text{out}} = 10^{2}$
- NFP tight: $_{\text{in}} = 10^{14}$, $_{\text{out}} = 10^{6}$

Multiple local minima / Knittel and Metaxoglou (2008)

- We find NFP with tight inner loop often finds global minimum
  - Multiple local minima do exist, but not insurmountable
- They used NFP and 50 starting values
- They claim BLP unreliable because different starting values find different local optima
-

Our alternative constrained optimization approach

- MPEC (general idea from Su & Judd 2007)

$$
\begin{aligned}
\min_{\theta,\xi} \quad & g(\;)' \, W g(\;) \\
\text{subject to} \quad & s(\;;\;) = \text{sW5 n 4)Q0 0 Td 5[(s)]TJ/F16}
\end{aligned}
$$

MPEC advantages vs. NFP

- No nested contraction mapping
  - No numerical error from inner loop
- Can be faster
  - Contraction mapping converges linearly vs. Newton's method (MPEC) converges quadratically
  - Market share equations hold only at final solution, not at every iteration
  - Market share equations exposed to optimizer
  - Optimizer has gradient and sparsity pattern of constraints to exploit
  - Objectives, constraints less nonlinear in parameters
  - Larger, smoother, sparser problem can be easier than smaller, rougher, denser problem
- Can be applied to models where there is no contraction mapping
  - Uniqueness (Gandhi 2008)
  - No uniqueness?

Lipschitz constants for NFP contraction mapping

Parameter Scale		Std. Dev. of Shocks ξ		# of Markets T		Mean of Intercept $E\ \ \beta_i^0$	
Value	Lipschitz	Value	Lipschitz	Value	Lipschitz	Value	Lipschitz
0.01	0.985	0.1	0.808	25	0.860	-2	0.771
0.1	0.971	0.25	0.813	50	0.871	-1	0.871
0.50	0.887	0.5	0.832	100	0.888	0	0.936
0.75	0.865	1	0.871	200	0.888	1	0.971
1	0.871	2	0.934			2	0.988
1.5	0.911	5	0.972			3	0.996
2	0.938	20	0.984			4	0.998
3	0.970						
5	0.993						

Speeds, # convergences and finite-sample performance

Intercept	Lips.	Routine	Runs	CPU	Own-Price Elasticities	
$E \quad \beta_i^0$	Const.		Conv.	Times	Bias	RMSE
-2	0.806	NFP tight	1	1184.1	0.026	0.254
		MPEC	1	1455.1	0.026	0.254
-1	0.895	NFP tight	1	1252.8	0.029	0.258
		MPEC	1	1528.4	0.029	0.258
0	0.950	NFP tight	1	1352.5	0.029	0.265
		MPEC	1	1564.1	0.029	0.265
1	0.978	NFP tight	1	1641.1	0.029	0.270
		MPEC	1	1562.5	0.029	0.270
2	0.991	NFP tight	1	2498.1	0.030	0.273
		MPEC	1	1480.7	0.030	0.273
3	0.997	NFP tight	1	5128.1	0.031	0.276
		MPEC	1	1653.9	0.030	0.278
4	0.999	NFP tight	1	9248.5	0.032	0.279
		MPEC	1	1881.8	0.031	0.279

Lessons learned

- For low Lipschitz constant, NFP and MPEC can be about the same speed
-

Field data: Nevo's cereal data

- NFP finds same local minimum for all 50 runs with objective function 4.5615
- MPEC finds same local minimum for 48 of 50 runs with objective function 4.5615
- Avg. CPU time: 763.14 sec (NFP) vs. 544 sec (MPEC)

Extension: Dynamic BLP with forward-looking consumers

- Consumers have expectations over future
  - Real option value of no-purchase: delay choice to future
    - Durable goods with declining prices
    - Stockpiling with temporary discounts
    - Purchasing upgrades and resale of existing products
  - Melnikov (2002), Nair (2007), Gowrisankaran and Rysman (2007), etc.
- Still endogeneity / stochastic model motivations for demand shocks $_{j,t}$

Example: durable goods with falling prices

- $J = 2$ products, R consumer types, T time periods

$$\log\left(p_{j,t}\right) = p_t^0 \quad {}_1 \quad {}_j + \quad {}_{j,t}$$

- Expected Value of waiting

$$v_0^r\left(p_t;\theta^r\right) = \delta \quad \max \left(\max_j \left\{ \begin{array}{l} v_0^r \left(p_t^0 \rho_j + \psi; \theta^r \right) + \epsilon_0 \\ \beta_j^r \quad \alpha^r \quad p_t^0 \rho_j + \psi \quad + \xi_j + \epsilon_j \end{array} \right\} \right) dF(\epsilon) dF_{\ ;}\ (\psi,\xi)$$

- Tastes ^{h}

$$^h_h = \begin{cases} {}^1; & \Pr(1) = {}_1 \\ \vdots & \vdots \\ {}^R; & \Pr(R) = 1 \quad \sum_{r=1}^{R \quad 1} {}_r \end{cases}$$

- Joint density of $\left({}_{j,t}; \ {}_{j,t} \right) \quad N\left(0; \quad \right)$

An MPEC approach to dynamic demand

Optimization problem

$$
\begin{array}{rl}
\max\limits_{\{\theta, \xi, \psi, \rho; v\}} & \prod\limits_{t=1} \frac{1}{(2\pi)^{\frac{3J}{2}} |\Omega|^{\frac{1}{2}}} exp \left(-\frac{1}{2} u_t' \Omega_u^{-1} u_t \right) |J_{t; u \rightarrow Y}| \\
\text{subject to} & s(\xi_t; \theta) = S_t \ \forall t = 1, \ldots, T \\
\text{and} & v_0^r(p_d) = \delta \log \left(\sum\limits_j \begin{array}{l} \exp\left(v_0^r(p_d^0 \rho_j + \psi)\right) + \ldots \\ \exp\left(\beta_j^r - \alpha^r (p_d^0 \rho_j + \psi) + \xi_j\right) \end{array} \right) dF_{\psi, \xi}(\psi, \xi) \\
& \forall d \in D, \ r = 1, \ldots, R.
\end{array}
$$

Constrained optimization combines

- Maximization of likelihood
- Dynamic programming
- Market share inversion / demand shocks

Early results from a Monte Carlo study

	Bias		RMSE	
θ	MPEC	NFP	MPEC	NFP
β_1 : 4	7.5E-03	4.6E-02	1.7E-01	1.5E-01
β_2 : -1	6.2E-03	3.7E-02	1.5E-01	1.2E-01
α : -0.15	-1.1E-04	-2.9E-04	8.0E-04	5.4E-04
ρ				
int_1 : 5	9.4E-03	1.9E-02	4.9E-02	4.6E-02
$\rho_{1;1}$: 0.8	9.5E-05	-2.1E-04	1.2E-03	1.2E-03
$\rho_{1;2}$: 0.2	-1.6E-04	-3.8E-05	1.5E-03	1.7E-03
int_2 : 5	8.9E-03	6.6E-04	5.9E-02	3.2E-02
$\rho_{2;1}$: 0.1	-7.0E-05	1.5E-04	1.1E-03	5.6E-04
$\rho_{2;2}$: 0.55	-6.5E-05	-4.5E-04	1.4E-03	8.8E-04
chol( )				
1	-4.1E-03	-4.5E-03	1.7E-02	1.7E-02
0.866	-1.7E-03	-5.5E-04	1.5E-02	1.4E-02
0.5	-7.9E-04	-2.4E-03	2.0E-02	1.9E-02
Avg CPU time (sec)	4579	16,971	4579	16,971

Conclusions

- BLP very important innovation in demand estimation
- Concerns with NFP algorithm
  - Can be slow
  - Numerical derivatives + loose inner loop can lead to incorrect parameter estimates
- MPEC applied to BLP
  - Can be faster
    - Especially when NFP's Lipschitz constant close to 1
  - Fewer numerical errors
    - No inner loop to propagate errors
  - Can apply to models where there is no contraction mapping
- Degree of advantage of MPEC over NFP may increase with dynamic BLP
  - NFP nests multiple inner loops
  - Typically linearly convergent contraction mappings
  - Amplifies benefits of quadratic convergence in MPEC