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

$$u_{i,j,t} = \beta_i^0 + x_{j,t}' \beta_i^x - \beta_i^p p_{j,t} + \xi_{j,t} + \varepsilon_{i,j,t}$$

- Consumer i, choice $j \in J$, market $t \in T$
- Product characteristics $x_{j,t}$, $p_{j,t}$, $\xi_{j,t}$
  - $\xi_{j,t}$ not in data
- β_i^0, β_i^x, β_i^p random coefficients
  - Distribution $F_\beta(\beta;\theta)$
  - BLP's statistical goal: estimate θ in parametric distribution
- $\epsilon_{i,j,t}$ extreme value shock (logit)
- i picks j if $u_{i,j,t} \geq u_{i,k,t} \forall\, k \in J, k \neq j$

Market shares

$$s_j\left(x_t, p_t, \xi_t; \theta\right) = \int_{\left\{\beta_i, \varepsilon_i \mid u_{i,j} \geq u_{i,j'}\ \forall j' \neq j\right\}} dF_\beta\left(\beta; \theta\right) dF_\epsilon\left(\varepsilon\right)$$

With logit errors

$$s_j\left(x_t, p_t, \xi_t; \theta\right) = \int_\beta \frac{\exp\left(\beta^0 + x'_{j,t}\beta^x - \beta^p p_{j,t} + \xi_{j,t}\right)}{1 + \sum_{k=1}^J \exp\left(\beta^0 + x'_{k,t}\beta^x - \beta^p p_{k,t} + \xi_{k,t}\right)} dF_\beta\left(\beta; \theta\right)$$

Simulate numerical integral

$$\hat{s}_j\left(x_t, p_t, \xi_t; \theta\right) = \frac{1}{ns}\sum_{r=1}^{ns} \frac{\exp\left(\beta^{0,r} + x'_{j,t}\beta^{x,r} - \beta^{p,r} p_{j,t} + \xi_{j,t}\right)}{1 + \sum_{k=1}^J \exp\left(\beta^{0,r} + x'_{k,t}\beta^{x,r} - \beta^{p,r} p_{k,t} + \xi_{k,t}\right)}$$

Inner loop of NFP approach

- Compute ξ numerically

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

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

$$\xi_t^{h+1} = \xi_t^h + \log S_t - \log s\left(\xi_t^h;\theta\right), \quad t = 1,\ldots,T$$

  - Stop when $\left\|\xi_t^h - \xi_t^{h+1}\right\| \leq \epsilon_{\text{in}}$
- Guaranteed to find a solution from any starting value
  - Why other nested methods (Newton's method, etc) not popular
  - Davis (2007)

Moment Condition

- Assume $E\left[\xi_{j,t} z_{j,t}\right] = 0$ for some instruments z_{jt}
  - Empirical analog

$$g\left(\xi\left(\theta\right)\right) = \frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{J}\xi_{j,t}\left(\theta\right)' z_{j,t} = \frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{J} s_{j,t}^{-1}\left(S;\theta\right)' z_{j,t}$$

- Data $\left\{\left(x_{j,t}, p_{j,t}, s_{j,t}, z_{j,t}\right)_{j=1}^{J}\right\}_{t=1}^{T}$

Two Approaches to GMM Estimator

- NFP: Inner loop

$$\min_{\theta} g\left(s^{-1}(S;\theta)\right)' W g\left(s^{-1}(S;\theta)\right)$$

  - For each guess of θ, need to find implied $\xi_{j,t}(\theta)$

- MPEC: Constrained optimization

$$\begin{array}{rl} \min\limits_{\theta,\xi} & g(\xi)' W g(\xi) \\ \text{subject to} & s(\xi;\theta) = S \end{array}$$

- Assume that $\mathcal{T}$ is a contraction mapping:

$$\left\|\mathcal{T}(\xi)-\mathcal{T}(\tilde{\xi})\right\| \leq L(\theta)\left\|\xi-\tilde{\xi}\right\|$$

  - $L < 1$ is called a Lipschitz constant

- The multidimensional equation $\xi = \mathcal{T}(\xi)$ has a unique solution ξ^*
  - Solution can be obtained by the convergent iteration process $\xi^{h+1} = \mathcal{T}(\xi^h)$, for $h = 0, 1, \ldots$
  - Convergence from "any" starting value.

- **The error at the h^{th} iteration is bounded**

$$\left\|\xi^h-\xi^*\right\| \leq \left\|\xi^h-\xi^{h-1}\right\| \frac{L(\theta)}{1-L(\theta)} \leq \left\|\xi^1-\xi^0\right\| \frac{L(\theta)^h}{1-L(\theta)}$$

Lipschitz constant for BLP contraction mapping

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

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

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

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

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

Theorems in Paper

Primarily using Taylor series / local analysis

- Use $\xi(\theta, \epsilon_{\text{in}})$, programmed objective function with nonzero tolerance and $\xi(\theta, 0)$, hypothetical inner loop with no numerical error
- Gradient, objective function numerical error is $O\left(\frac{L(\theta)}{1-L(\theta)}\epsilon_{\text{in}}\right)$
- For the outer-loop GMM minimization to converge, the outer-loop tolerance ϵ_{out} should be chosen to satisfy $\epsilon_{\text{out}} = O\left(\frac{L(\theta)}{1-L(\theta)}\epsilon_{\text{in}}\right)$
- If $\epsilon_{\text{in}} = 10^{-6}$, the numerical error in parameter estimates is around 10^{-3}
- Numerical error does not go away in large samples

Loose inner loop + numerical derivatives = bad news

Application of Lemma 9.1 in Nocedal & Wright (2006)

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

$$\nabla_d Q\left(\xi\left(\theta, \epsilon_{\text{in}}\right)\right) = \left\{ \frac{Q\left(\xi\left(\theta + de_k, \epsilon_{\text{in}}\right)\right) - Q\left(\xi\left(\theta - de_k, \epsilon_{\text{in}}\right)\right)}{2d} \right\}_{k=1}^{|\theta|}$$

- Gradient error bounded

$$\left\|\nabla_d Q\left(\xi\left(\theta, \epsilon_{\text{in}}\right)\right) - \nabla Q\left(\xi\left(\theta, 0\right)\right)\right\|_\infty \leq O\left(d^2\right) + \frac{1}{d} O\left(\frac{L\left(\theta\right)}{1 - L\left(\theta\right)} \epsilon_{\text{in}}\right)$$

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

Simulated data setup

- $T = 50$, $J = 25$ (large T needed to identify intercept)

$$\begin{bmatrix} x_{1,j,t} \\ x_{2,j,t} \\ x_{3,j,t} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & -0.8 & 0.3 \\ -0.8 & 1 & 0.3 \\ 0.3 & 0.3 & 1 \end{bmatrix} \right)$$

- $\xi_{j,t} \sim N(0,1)$
- $p_{j,t} = \left| 0.5 \cdot \xi_{j,t} + e_{j,t} + 1.1 \cdot \sum_{k=1}^{3} x_{k,jt,t} \right|$
- $z_{j,t,d} \sim U\left(\frac{1}{4} p_{j,t} - 0.5 \cdot \xi_{j,t}, 1\right)$, $D = 6$ instruments
- $F_\beta(\beta; \theta)$: 5 independent normal distributions ($K = 3$ attributes, price and the intercept)
- $\beta_i = \left\{ \beta_i^0, \beta_i^1, \beta_i^2, \beta_i^3, \beta_i^p \right\}$: $E[\beta_i] = \{0.1, 1.5, 1.5, 0.5, -3\}$ and $\mathrm{Var}[\beta_\mathrm{i}] = \{0.5, 0.5, 0.5, 0.5, 0.2\}$

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
  - Professional quality optimization program
  - Can be called directly from R2008a version of MATLAB
  - We call from TOMLAB

Loose versus tight tolerances for NFP

With numerical derivatives

	NFP Loose Inner	NFP Loose Both	NFP Tight	Truth
Fraction Convergence	0.0	0.54	0.95	
Frac.$<1\%>$ "Global" Min.	0.0	0.0	1.00	
Mean Own Price Elasticity	-7.24	-7.49	-5.77	-5.68
Std. Dev. Own Price Elasticity	5.48	5.55	~0	
Lowest Objective	0.0176	0.0198	0.0169	
Elasticity for Lowest Obj.	-5.76	-5.73	-5.77	-5.68

- 100 starting values for one dataset
- NFP loose inner loop: $\epsilon_{\text{in}} = 10^{-4}$, $\epsilon_{\text{out}} = 10^{-6}$
- NFP loose both: $\epsilon_{\text{in}} = 10^{-4}$, $\epsilon_{\text{out}} = 10^{-2}$
- NFP tight: $\epsilon_{\text{in}} = 10^{-14}$, $\epsilon_{\text{out}} = 10^{-6}$

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: $\epsilon_{\text{in}} = 10^{-4}$, $\epsilon_{\text{out}} = 10^{-6}$
- NFP loose both: $\epsilon_{\text{in}} = 10^{-4}$, $\epsilon_{\text{out}} = 10^{-2}$
- NFP tight: $\epsilon_{\text{in}} = 10^{-14}$, $\epsilon_{\text{out}} = 10^{-6}$

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
- We find they did not check solver error messages, used unreliable solvers

Lessons learned

- Loose inner loop causes numerical error in gradient
  - Can converge to wrong point
  - Can fail to diagnose convergence of outer loop
  - Early stops, false estimates
- Making outer loop tolerance loose allows "convergence"
  - But to false solution
- We will now code derivatives
  - Improves performance of smooth optimizers
  - Same component functions for derivatives
  - Helpful for standard errors

Our alternative constrained optimization approach

- MPEC (general idea from Su & Judd 2007)

$$\begin{array}{rl} \min\limits_{\theta,\xi} & g(\xi)' W g(\xi) \\ \text{subject to} & s(\xi;\theta) = S \end{array}$$

- Market share equations enter as constraints
- No inner loop / contraction mapping
- MPEC uses constrained optimization: standard numerical problem
- The moment condition term $g(\xi)$ is just

$$g(\xi) = \frac{1}{T}\sum_{t=1}^{T} \xi_t z_t$$

MPEC and NFP have identical local minima

Theorem

Set of first order conditions to MPEC problem equivalent to set of first order conditions to true (no numerical error) NFP

Proof

- NFP is $\min_{\theta} Q\left(\xi\left(\theta\right)\right)$
  - FOC is $\frac{\partial Q(\xi(\theta))}{\partial \theta} = \frac{d\xi}{d\theta}' \frac{\partial Q}{\partial \xi} = 0$
- MPEC Lagrangian is $\mathcal{L}\left(\theta, \xi, \lambda\right) = Q(\xi) - \lambda^T \left(S - s(\xi; \theta)\right)$
  - FOC's include $\dfrac{\partial \mathcal{L}\left(\theta, \xi, \lambda\right)}{\partial \theta} = -\dfrac{ds(\xi;\theta)}{d\theta}' \lambda = 0$
  - $\dfrac{\partial \mathcal{L}\left(\theta, \xi, \lambda\right)}{\partial \xi} = \dfrac{\partial Q}{\partial \xi} - \dfrac{ds(\xi;\theta)}{d\xi}' \lambda = 0$
- $\frac{\partial \mathcal{L}}{\partial \theta} = -\dfrac{ds(\xi;\theta)}{d\theta}' \left(\dfrac{ds(\xi;\theta)}{d\xi}'\right)^{-1} \dfrac{\partial Q}{\partial \xi} = 0$, ($\frac{ds(\xi;\theta)}{d\xi}$ is invertible)
- Implicit function theorem: $\frac{\partial \xi(\theta)}{\partial \theta} = -\left(\frac{ds(\xi;\theta)}{d\xi}\right)^{-1} \frac{ds(\xi;\theta)}{d\theta}$

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 $E\left[\beta_i^0\right]$	
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	
$E\left[\beta_i^0\right]$	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
- For high Lipschitz constant, NFP may become very slow
  - Remember: multiply times by 100 for reasonable simulation draws!
- MPEC speed can be relatively invariant to Lipschitz constant
  - No contraction mapping in MPEC

Speed for varying # of markets

- Concern: MPEC has lots of auxiliary optimization parameters (# of markets times # of products)
  - MPEC has trade-off between quadratic convergence and dimension of optimization
  - MPEC may perform poorly with large numbers of markets
- Answer: MPEC may be competitive with NFP in these settings

# Mark. T	Lips. Const.	Routine	Runs Conv.	CPU Time
25	0.937	NFP	1	258.5
		MPEC	1	226.8
50	0.944	NFP	1	780.0
		MPEC	1	564.7
100	0.951	NFP	1	2559.6
		MPEC	1	2866.0
200	0.953	NFP	1	6481.7
		MPEC	1	2543.6

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: maximum likelihood estimation of BLP model

- MLE more efficient, facilitates testing models
  - Add parametric distributional assumption on $\xi_{j,t}$
- Still need NFP / MPEC
- Likelihood has Jacobian term from $\xi_{j,t}$'s to shares

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 $\xi_{j,t}$

Example: durable goods with falling prices

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

$$\log\left(p_{j,t}\right) = p'_{t-1}\rho_j + \psi_{j,t}$$

- Expected Value of waiting

$$v_0^r\left(p_t;\theta^r\right) = \delta \int \max\left\{ \begin{array}{c} v_0^r\left(p'_t\rho_j + \psi;\theta^r\right) + \epsilon_0 \\ \max_j \left\{\beta_j^r - \alpha^r\left(p'_t\rho_j + \psi\right) + \xi_j + \epsilon_j\right\} \end{array} \right\} dF_\epsilon(\epsilon) dF_{\psi,\xi}\left(\psi,\xi\right)$$

- Tastes $\theta^h \equiv \left[\begin{array}{c} \beta^h \\ \alpha^h \end{array}\right] = \begin{cases} \theta^1, & \Pr(1) = \lambda_1 \\ \vdots & \vdots \\ \theta^R, & \Pr(R) = 1 - \sum_{r=1}^{R-1}\lambda_r \end{cases}$

- Joint density of $\left(\xi_{j,t}, \eta_{j,t}\right) \sim N\left(0,\Omega\right)$

Empirical model

$$u_t \equiv \begin{bmatrix} \psi_t \\ \xi_t \end{bmatrix} = \begin{bmatrix} \log\left(p_{j,t}\right) - p'_{t-1}\rho_j \\ s^{-1}\left(p_t, S_t; \theta\right) \end{bmatrix}$$

$$s_j\left(p_t;\theta\right) = \sum_{r=1}^{R} \lambda_{t,r} \frac{\exp(\beta_j^r - \alpha^r p_{j,t} + \xi_{j,t})}{\exp\left(v_0^r(p_t;\theta^r)\right) + \sum_{k=1}^{J} \exp\left(\beta_k^r - \alpha^r p_{k,t} + \xi_{k,t}\right)}$$

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

Optimization problem

$$\max_{\{\theta,\rho,\Omega\}} \prod_{t=1}^{T} \frac{1}{(2\pi)^{\frac{3J}{2}} \left|\Omega\right|^{\frac{1}{2}}} \exp\left(-\frac{1}{2} u_t^{'} \Omega_u^{-1} u_t\right) \left|J_{t,u\rightarrow Y}\right|$$

NFP algorithm now has three loops!

- Two inner loops
  - Compute consumer's value function, $v_0^r(p_t)$ (Bellman equation is a contraction mapping)
  - Compute $\xi_{j,t}$ by inverting market shares (BLP contraction mapping)
  - In some dynamic models, BLP contraction mapping may not converge
- Outer loop is optimization of the objective function (max of likelihood)
- Exacerbates incentive to loosen tolerances
- Some specifications may have additional inner loops
  - Complementary goods' demand shocks (Lee 2008)
  - Value function after buying (Gowrisankaran and Rysman 2007, Dube, Hitsch and Chintagunta 2008)

An MPEC approach to dynamic demand

Optimization problem

$$\max_{\{\theta,\rho,\Omega,\xi,v\}} \prod_{t=1}^{T} \frac{1}{(2\pi)^{\frac{3J}{2}} |\Omega|^{\frac{1}{2}}} exp\left(-\frac{1}{2} u_t^{'} \Omega_u^{-1} u_t\right) |J_{t,u\rightarrow Y}|$$

subject to $s(\xi_t;\theta) = S_t \; \forall \, t = 1, \ldots, T$

and $$v_0^r\left(p_d\right) = \delta \log \left(\begin{array}{c} \exp\left(v_0^r(p_d'\rho_j + \psi)\right) + \ldots \\ \sum_j \exp\left(\beta_j^r - \alpha^r\left(p_d'\rho_j + \psi\right) + \xi_j\right) \end{array} \right) dF_{\psi,\xi}\left(\psi,\xi\right)$$

$\forall \, d \in D, \; r = 1, \ldots, R.$

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
$\mathrm{chol}(\Omega)$				
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