Best Practices for Demand Estimation with `pyblp`

Chris Conlon and Jeff Gormaker

NYU Stern and NY Fed

April 5, 2019

An Ongoing Project...

What do we have so far?

- Available on PyPI: `pip install pyblp`
- Extensive documentation: `https://pyblp.readthedocs.io/en/stable/`
- Long list of features
- 100+ active users. (hopefully more).

Today's Talk

Theory Part

- Slight re-writing of the usual BLP problem
- Focus on Simultaneous Supply+ Demand.
- Focus on optimal IV. (For intuition).

Numerics Part

- Fixed Point solutions: SQUAREM, Newton/ LM.
- Integration Methods: Quadrature/Sparse Grids
- Optimization Methods: Less sensitive than anticipated.
- Over/Under-flow is deadly.

A Famous Example

```
blp_problem = pyblp.Problem(
    product_formulations=(
        pyblp.Formulation('1 + hpwt + air + mpd + space'),                  # Linear demand
        pyblp.Formulation('1 + prices + hpwt + air + mpd + space'),         # Nonlinear demand
        pyblp.Formulation('1 + log(hpwt) + air + log(mpg) + log(space) + trend')  # Supply
    ),
    agent_formulation=pyblp.Formulation('0 + I(1 / income)'),               # Demographics
    product_data=pandas.read_csv(pyblp.data.BLP_PRODUCTS_LOCATION),
    agent_data=pandas.read_csv(pyblp.data.BLP_AGENTS_LOCATION)
)
print(blp_problem)
```

```
Dimensions:
============================================
 N     T    K1    K2    K3    D    MD    MS
----  ---  ----  ----  ----  ---  ----  ----
2217  20    5     6     6     1    11    12
============================================

Formulations:
=====================================================================================
       Column Indices:            0          1         2       3         4        5
-----------------------------  --------  ---------  -----  --------  ----------  -----
 X1: Linear Characteristics       1        hpwt      air     mpd       space
X2: Nonlinear Characteristics     1       prices     hpwt    air        mpd      space
  X3: Cost Characteristics        1      log(hpwt)   air   log(mpg)  log(space)  trend
       d: Demographics         1/income
=====================================================================================
```

Another Famous Example

```
nevo_problem = pyblp.Problem(
    product_formulations=(
        pyblp.Formulation('0 + prices', absorb='C(product_ids)'),                     # Linear demand
        pyblp.Formulation('1 + prices + sugar + mushy')                               # Nonlinear demand
    ),
    agent_formulation=pyblp.Formulation('0 + income + income_squared + age + child'),  # Demographics
    product_data=pandas.read_csv(pyblp.data.NEVO_PRODUCTS_LOCATION),
    agent_data=pandas.read_csv(pyblp.data.NEVO_AGENTS_LOCATION)
)
print(nevo_problem)
```

```
Dimensions:
======================================
 N     T    K1    K2    D    MD    ED
----  ---  ----  ----  ---  ----  ----
2256  94    1     4     4    20    1
======================================

Formulations:
===================================================================
       Column Indices:           0            1           2      3
-----------------------------  ------  --------------  -----  -----
 X1: Linear Characteristics    prices
X2: Nonlinear Characteristics    1         prices      sugar  mushy
       d: Demographics         income  income_squared   age   child
===================================================================
```

Tons of Features

In one line you can:

- Estimate the model
- Compute Optimal instruments
- Caclulate elastictities, diversion ratios
- Calculate implied *MC*
- Solve for pricing equilibria
- Compute merger effects

More at: `https://pyblp.readthedocs.io/en/stable/`

Theory Part

Some setup

We can break up the parameter space into three parts:

- θ_1: linear exogenous demand parameters,
- θ_2: nonlinear endogenous parameters including price and random coefficients
- θ_3: linear exogenous supply parameters.

(a) For each market t: solve $\mathcal{S}_{jt} = s_{jt}(\delta_{\cdot t}, \theta_2)$ for $\widehat{\delta}_{\cdot t}(\theta_2)$.

(b) For each market t: use $\widehat{\delta}_{\cdot t}(\theta_2)$ to construct $\eta_{\cdot t}(\mathbf{q_t}, \mathbf{p_t}, \widehat{\delta}_{\cdot t}(\theta_2), \theta_2)$

(c) For each market t: Recover $\widehat{mc}_{jt}(\widehat{\delta}_{\cdot t}(\theta_2), \theta_2) = p_{jt} - \eta_{jt}(\widehat{\delta}_{\cdot t}(\theta_2), \theta_2)$

(d) Stack up $\widehat{\delta}_{\cdot t}(\theta_2)$ and $\widehat{mc}_{jt}(\widehat{\delta}_{\cdot t}(\theta_2), \theta_2)$ and use linear IV-GMM to recover $[\widehat{\theta}_1(\theta_2), \widehat{\theta}_3(\theta_2)]$ following the recipe in Appendix

(e) Construct the residuals:

$$\widehat{\xi}_{jt}(\theta_2) = \widehat{\delta}_{jt}(\theta_2) - x_{jt}\widehat{\beta}(\theta_2) + \alpha p_{jt}$$
$$\widehat{\omega}_{jt}(\theta_2) = \widehat{mc}_{jt}(\theta_2) - [x_{jt}\, w_{jt}]\,\widehat{\gamma}(\theta_2)$$

(f) Construct sample moments

$$g_n^D(\theta_2) = \frac{1}{N}\sum_{jt} Z_{jt}^{D\prime}\widehat{\xi}_{jt}(\theta_2)$$
$$g_n^S(\theta_2) = \frac{1}{N}\sum_{jt} Z_{jt}^{S\prime}\widehat{\omega}_{jt}(\theta_2)$$

(g) Construct GMM objective $Q_n(\theta_2) = \begin{bmatrix} g_n^d(\theta_2) \\ g_n^s(\theta_2) \end{bmatrix}' W \begin{bmatrix} g_n^d(\theta_2) \\ g_n^s(\theta_2) \end{bmatrix}$

Additional Details

Some different definitions:

$$\begin{aligned} y_{jt}^{D} &:= \widehat{\delta}_{jt}(\theta_2) + \alpha p_{jt} &= (x_{jt}\ \sigma_{jt})\beta + \xi_t =: x_{jt}^{D\prime}\beta + \xi_{jt} \\ y_{jt}^{S} &:= p_{jt} - \widehat{\eta}_{jt}(\theta_2) &= (x_{jt}\ w_{jt})'\gamma + \omega_t =: x_{jt}^{S\prime}\gamma + \omega_{jt} \end{aligned} \tag{1}$$

Stacking the system across observations yields:[1]

$$\underbrace{\begin{bmatrix} y_D \\ y_S \end{bmatrix}}_{2N\times 1} = \underbrace{\begin{bmatrix} X_D & 0 \\ 0 & X_S \end{bmatrix}}_{2N\times(K_1+K_3)} \underbrace{\begin{bmatrix} \beta \\ \gamma \end{bmatrix}}_{(K_1+K_3)\times 1} + \underbrace{\begin{bmatrix} \xi \\ \omega \end{bmatrix}}_{2N\times 1} \tag{2}$$

Because we got rid of endogeneity (on LHS) we can now incorporate high dimensional FE!

[1] Note: we cannot perform independent regressions unless we are willing to assume that $Cov(\xi_{jt}, \omega_{jt}) = 0$.

Optimal Instruments

How to construct optimal instruments in form of Chamberlain (1987)

$$E\left[\frac{\partial \xi_{jt}}{\partial \theta}|X_t, w_{jt}\right] = \left[\beta, E\left[\frac{\partial \xi_{jt}}{\partial \alpha}|X_t, w_{jt}\right], E[\frac{\partial \xi_{jt}}{\partial \sigma}|X_t, w_{jt}]\right]$$

Some challenges:

1. p_{jt} depends on X_t, w_t, ξ_t in a highly nonlinear way (no explicit solution!).
2. $E[\frac{\partial \xi_{jt}}{\partial \sigma}|X_t, w_t] = E[[\frac{\partial \mathbf{s_t}}{\partial \delta_\mathbf{t}}]^{-1}[\frac{\partial \mathbf{s_t}}{\partial \sigma}]|X_t, w_t]$ (not conditioned on endogenous p!)

"Feasible" Recipe:

1. Fix $\hat{\theta} = (\hat{\alpha}, \hat{\beta}, \hat{\sigma})$ and draw ξ_t from empirical density
2. Solve fixed point equation for $\hat{p_{jt}}$
3. Compute necessary Jacobian
4. Average over all values of ξ_t. (Lazy approach: use only $\xi = 0$).

Chamberlain (1987) tells us the optimal instruments for this supply-demand system of $G\Omega^{-1}$ where for a given observation n,

$$G_n := \underbrace{\begin{bmatrix} \frac{\partial \xi}{\partial \beta} & \frac{\partial \omega}{\partial \beta} \\ \frac{\partial \xi}{\partial \alpha} & \frac{\partial \omega}{\partial \alpha} \\ \frac{\partial \xi}{\partial \sigma} & \frac{\partial \omega}{\partial \sigma} \\ \frac{\partial \xi}{\partial \gamma} & \frac{\partial \omega}{\partial \gamma} \end{bmatrix}_n}_{(K_1+K_2+K_3)\times 2} = \begin{bmatrix} -x_j & 0 \\ -v_j & 0 \\ \xi_\alpha & \omega_\alpha \\ \xi_\sigma & \omega_\sigma \\ 0 & -x_j \\ 0 & -w_j \end{bmatrix}_n \qquad \Omega := \underbrace{\begin{bmatrix} \sigma_\xi^2 & \sigma_{\xi\omega} \\ \sigma_{\xi\omega} & \sigma_\omega^2 \end{bmatrix}}_{2\times 2}$$

Optimal Instruments

$$G_n\Omega^{-1} = \frac{1}{\sigma_\xi^2\sigma_\omega^2 - (\sigma_{\xi\omega})^2} \times \begin{bmatrix} -\sigma_\omega^2 X & \sigma_{\xi\omega} X \\ -\sigma_\omega^2 V & \sigma_{\xi\omega} V \\ \sigma_\omega^2 \xi_\alpha - \sigma_{\xi\omega}\omega_\alpha & \sigma_\xi^2 \omega_\alpha - \sigma_{\xi\omega}\xi_\alpha \\ \sigma_\omega^2 \xi_\sigma - \sigma_{\xi\omega}\omega_\sigma & \sigma_\xi^2 \omega_\sigma - \sigma_{\xi\omega}\xi_\sigma \\ \sigma_{\xi\omega} X & -\sigma_\xi^2 X \\ \sigma_{\xi\omega} W & -\sigma_\xi^2 W \end{bmatrix}_n$$

Clearly rows 1 and 5 are co-linear.

Optimal Instruments

$$(G_n\Omega^{-1}) \circ \Theta = \frac{1}{\sigma_\xi^2\sigma_\omega^2 - (\sigma_{\xi\omega})^2} \times \begin{bmatrix} -W_\omega^2 X & 0 \\ -\sigma_\omega^2 V & \sigma_{\xi\omega} V \\ \sigma_\omega^2 \xi_\alpha - \sigma_{\xi\omega}\omega_\alpha & \sigma_\xi^2 \omega_\alpha - \sigma_{\xi\omega}\xi_\alpha \\ \sigma_\omega^2 \xi_\sigma - \sigma_{\xi\omega}\omega_\sigma & \sigma_\xi^2 \omega_\sigma - \sigma_{\xi\omega}\xi_\sigma \\ 0 & -\sigma_\xi^2 X \\ \sigma_{\xi\omega} W & -\sigma_\xi^2 W \end{bmatrix}_n$$

Now we can partition our instrument set by column into “demand” and “supply” instruments as

$$z_{nD} := (G_n\Omega^{-1} \circ \Theta)_{\cdot 1}$$
$$z_{nS} := (G_n\Omega^{-1} \circ \Theta)_{\cdot 2}$$

Aside: What does Supply tell us about Demand?

Under optimal IV these are overidentifying restrictions

- Cross-equation restrictions for endogenous parameters K_2 of them
- Exclusion Restrictions from v_{jt} (demand shifters) and w_{jt} (cost shifters.) $K_1 + K_3 - 2K_x$.
- Overall: $K - K_x$ overidentifying restrictions from supply.
- Can test: supply model, conduct, etc.

Numerical Improvements

Stability

There are a few places where estimation usually goes wrong

- Problem is basically linear except for the contraction step.
- Computation of shares has both underflow/overflow possibilities
  - Shares $\rightarrow$ 0 (underflow)
  - Shares $\rightarrow$ 1 (overflow)
- Happens with either large values of σ or large draws of ν_i.
  - Bounding parameters helps a lot.
- Additional numerical problem with $\sum_j e^{\sigma_{ij}}$.
  - Summing large numbers of small numbers and one large number can lead to loss of precision.
  - the usual trick: $\frac{e^{\sigma_{ij}}}{\sum_k e^{\sigma_{ik}}} = \frac{e^{\sigma_{ij}-m}}{\sum_k e^{\sigma_{ik}-m}}$ where $m = \max_k \sigma_{ik}$ or $m = \min_k \sigma_{ik}$.
  - Python interpreter seems to be a bit better about avoiding loss of precision at runtime.

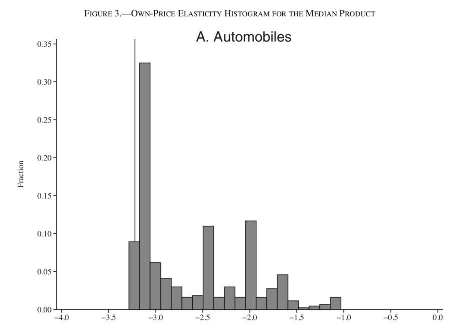
Optimization: Existing Literature

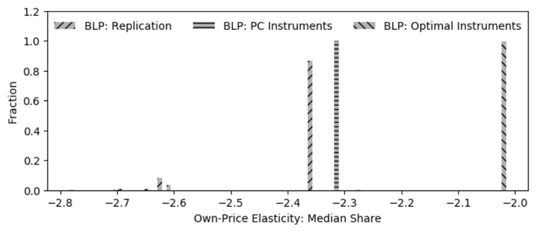
- Knittel Metaxoglou (2014): download Nevo (2000)'s code, download 50 solvers for MATLAB, and find about 60 different "local minima".
- Dube, Fox, Su (2012): show how to fix some bugs in the gradient, and show that MPEC approach appears to be more reliable in finding the "global minimum".
- Main takeaway is probably that choice of optimizer matters and we should use multiple starting values to make sure we have found the "global minimum".

Our Findings

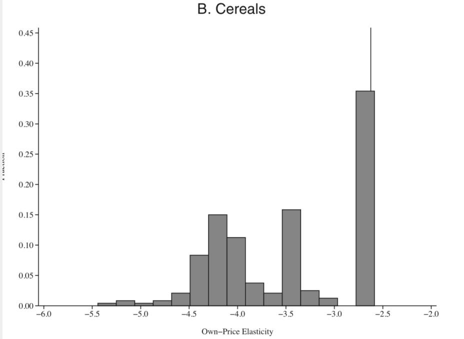
- Nearly all optimizers work reasonably well (around 99%) of time.
  - Gradient within tolerance of zero (element by element)
  - Hessian is PSD (all eigenvalues non-negative).
- Almost no discernible difference in recovered parameters.
- Even Nelder-Mead(!) does surprisingly well
- Perhaps our problems are too easy?

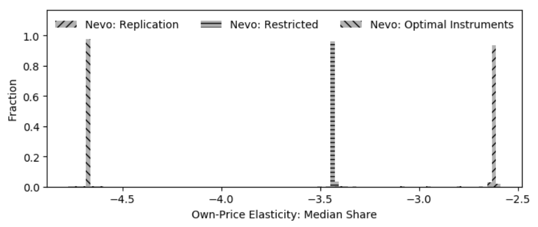
Knittel Metaxoglou (2014) vs. pyblp : Autos





Knittel Metaxoglou (2014) vs. pyblp: RTE Cereal





Monte Carlo Setup

Monte Carlo Setup

- 1000 simulated datasets
- $T = 20$ markets, and in each market, the number of firms is chosen randomly from $F_t \in \{F-2, F-1, F\} = \{3, 4, 5\}$.
- Each firm produces a number of products chosen randomly from $J_f \in \{2, 5, 10\}$.
- Sample sizes are generally between $N = 200$ and $N = 600$.
- Unobserved product characteristics, ξ and ω, are drawn from a mean-zero bivariate normal distribution with $\text{Var}(\xi) = \text{Var}(\omega) = 0.1$ and $\text{Corr}(\xi, \omega) = 0.5$.
- ν, consist of 1,000 draws from the standard normal distribution, and differ across markets.

Monte Carlo Setup

- Observed linear product characteristics are $X_1 = [1, x, p]$, with a single random coeffcient $X_2 = x$
- Linear supply characteristics $X_3 = [1, x, w]$. Exogenous characteristics,
- (x, w), are drawn from the standard uniform distribution.
- Endogenous Bertrand-Nash prices and shares $(\mathbf{p}, \mathbf{s})$ are computed via iteration over the ζ-markup equation.
- Demand-side parameters are $(\beta_0, \beta_1, \alpha) = [-7, 6, -1]'$ and $\sigma_x = 3$.
- Other linear parameters were chosen to generate realistic outside shares that are generally between $s_{0t} = 0.8$ and $s_{0t} = 0.9$.
- Supply-side parameters are $(\gamma_0, \gamma_1, \gamma_2) = [2, 1, 1]'$.

Simple 1 random coefficient on x.

Complicated adds a random coefficient on price: $X_2 = (x, p)$, $\Sigma_{22} = 0.2$, and $\Sigma_{12} = \Sigma_{21} = 0$.

RCNL adds a nesting parameter: $\rho = 0.5$ and each of the J_f products produced by a firm is randomly assigned to one of $H = 2$ nesting groups.

BLP has simultaneous supply and demand, an an interaction between income and price.

Nevo has diagonal Σ and demographics Π.

Best Practices

Best Practices

1. **Speed Up Contraction**: `SQUAREM` (Reynaerts, Varodayan, Nash 2014), Newton-type Methods (Levenberg/Marquardt).
2. **Reduce Integration Error**: `sparse-grids` (Heiss and Winchel 2007)
3. **Solve Pricing Equilibria**: Modified ζ fixed point of Morrow and Skerlos (2010)
4. **Optimal Instruments**: Using supply and demand restrictions ("approximate" version)

BLP also propose a fixed-point approach to solve the $J_t \times J_t$ system of equations for shares. They show that the following is a contraction mapping $f(\delta) = \delta$:

$$f : \delta_{\cdot t}^{h+1} \leftarrow \delta_{\cdot t}^{h} + \ln \mathcal{S}_{\cdot t} - \ln \mathbf{s_{\cdot t}}(\delta_{\cdot t}^{h}, \theta_2) \tag{3}$$

- This kind of contraction mapping is linearly convergent where the rate of convergence is proportional to $\frac{L(\theta_2)}{1-L(\theta_2)}$ where $L(\theta_2)$ is the Lipschitz constant.
- Because (3) is a contraction, we know that $L(\theta_2) < 1$.
- DFS2012 show that for the BLP contraction the Lipschitz constant is defined as $L(\theta_2) = \max_{\delta \in \Delta} \left\| \mathbf{I}_{J_t} - \frac{\partial \log \mathbf{s_{\cdot t}}}{\partial \delta_{\cdot t}}(\delta_{\cdot t}, \theta_2) \right\|_\infty$.

#1: Accelerating the Contraction (Newton's Method)

$$\delta_{\cdot t}^{h+1} \leftarrow \delta_{\cdot t}^{h} - \lambda J_{\mathbf{s}}^{-1}(\delta_{\cdot t}^{h}, \theta_2) \cdot \mathbf{s_t}(\delta_{\cdot t}^{h}, \theta_2)$$

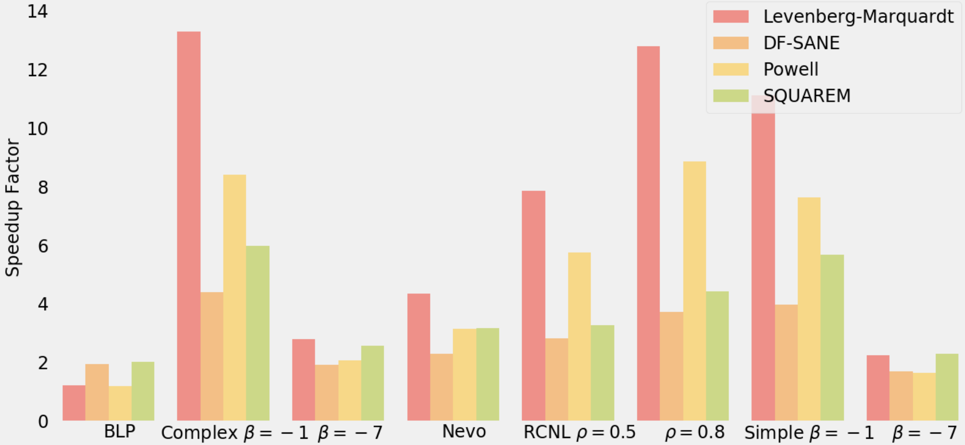
- Each Newton-Raphson iteration would require computation of:
  - J_t vector of marketshares $\mathbf{s_t}(\delta_{\cdot t}^{h}, \theta_2)$, the $J_t \times J_t$
  - Jacobian matrix $J_{\mathbf{s}}(\delta_{\cdot t}^{h}, \theta_2) = \frac{\partial \mathbf{s_{\cdot t}}}{\partial \delta_{\cdot t}}(\delta_{\cdot t}^{h}, \theta_2)$
  - as well as its inverse $J_{\mathbf{s}}^{-1}(\delta_{\cdot t}^{h})$.
- Inverse Jacobian can be costly when J_t is large (and requires integration).
- Try two routines Levenberg/Marquardt (good) and Powell's method (`hybrj`) less good.

#1: Accelerated Fixed Points

Most of these methods use information from mutliple iterations $(\delta^h, \delta^{h+1}, \delta^{h+2}, f(\delta^h), f(f(\delta^h)))$ to approximate $J_{\mathbf{s}}$ or $J_{\mathbf{s}}^{-1}$:

$$\begin{aligned} \delta_{\cdot t}^{h+1} &= \delta_{\cdot t}^{h} - 2\alpha^h r^h + (\alpha^h)^2 v^h, \quad \alpha^h = \frac{(v^h)' r^h}{(v^h)' v^h} \\ r^h &= f(\delta_{\cdot t}^h) - \delta_{\cdot t}^h, \quad v^h = f(f(\delta_{\cdot t}^h)) - 2f(\delta_{\cdot t}^h) + \delta_{\cdot t}^h \end{aligned} \tag{4}$$

- This particular algorithm is known as SQUAREM used in biostats for EM algorithms.
- Applied to BLP by Reynaerts, Varadhan, Nash (2012).
- Iterations are more costly but much more accurate (almost a Newton step).
- Speedup is 2-12x.



Levenberg-Marquardt
DF-SANE
Powell
SQUAREM
Speedup Factor
0
2
4
6
8
10
12
14
BLP
Complex $\beta = -1$
$\beta = -7$
Nevo
RCNL $\rho = 0.5$
$\rho = 0.8$
Simple $\beta = -1$
$\beta = -7$

#2: Numerical Integration

$$\hat{f} = \sum_i w_i \cdot f(\nu_i) \cdot g(\nu_i|\theta)$$

- Most of literature does Monte Carlo Integration
- quadrature rules are exact to polynomial order of approximation
  - Gauss-Hermite works very well when $g(\nu_i) \propto \exp[-x^2]$.
- but scale poorly in high dimensions: curse of dimensionality
- Heiss and Winschel: suggest sparse grids quadrature rules which drop nodes in clever ways
- Skrainka (2011) and Skrainka and Judd (2012) suggest monomial rules.

#2: Numerical Integration

					True Value				Median Bias				Median Absolute Error			
Simulation	Supply	Integration	I_t	Seconds	α	σ_x	σ_p	ρ	α	σ_x	σ_p	ρ	α	σ_x	σ_p	ρ
Simple	Yes	Monte Carlo	100	5.9	-1	3			0.104	-0.659			0.226	0.660		
Simple	Yes	Monte Carlo	1,000	38.1	-1	3			0.027	-0.090			0.178	0.186		
Simple	Yes	Halton	1,000	36.4	-1	3			0.021	0.036			0.171	0.152		
Simple	Yes	Product Rule	9^1	4.3	-1	3			0.010	0.011			0.171	0.159		
Complex	Yes	Monte Carlo	100	11.2	-1	3	0.2		0.120	-0.686	-0.135		0.277	0.750	0.153	
Complex	Yes	Monte Carlo	1,000	52.5	-1	3	0.2		0.015	-0.123	-0.049		0.205	0.234	0.128	
Complex	Yes	Halton	1,000	52.6	-1	3	0.2		-0.034	-0.017	0.032		0.194	0.169	0.141	
Complex	Yes	Product Rule	9^2	11.3	-1	3	0.2		-0.063	-0.041	0.044		0.185	0.179	0.136	
RCNL	Yes	Monte Carlo	100	30.8	-1	3		0.5	0.097	-0.614		0.037	0.129	0.615		0.046
RCNL	Yes	Monte Carlo	1,000	154.2	-1	3		0.5	0.029	-0.099		0.005	0.109	0.186		0.020
RCNL	Yes	Halton	1,000	160.7	-1	3		0.5	0.017	0.010		0.000	0.114	0.173		0.018
RCNL	Yes	Product Rule	9^1	20.5	-1	3		0.5	0.008	-0.013		0.001	0.108	0.171		0.018

Recall the multi-product Bertrand FOCs:

$$\begin{aligned}\arg\max_{p \in \mathcal{J}_f} \pi_f(\mathbf{p}) &= \sum_{j \in \mathcal{J}_f} (p_j - c_j) \cdot q_j(\mathbf{p}) \\ \rightarrow 0 &= q_j(\mathbf{p}) + \sum_{k \in \mathcal{J}_f} (p_k - c_k) \frac{\partial q_k}{\partial p_j}(\mathbf{p})\end{aligned}$$

It is helpful to define the matrix Ω with entries:

$$\Omega_{(j,k)}(\mathbf{p}) = \left\{ \begin{array}{cc} -\frac{\partial q_j}{\partial p_k}(\mathbf{p}) & \text{for } (j,k) \in \mathcal{J}_f \\ 0 & \text{for } (j,k) \notin \mathcal{J}_f \end{array} \right\}$$

We can re-write the FOC in matrix form:

$$q(\mathbf{p}) = \Omega(\mathbf{p}) \cdot (\mathbf{p} - \mathbf{mc})$$

- Can we iterate on the price relation until we converge to a new equilibrium?

$$\mathbf{p} \leftarrow \widehat{\mathbf{mc}} - \Omega(\mathbf{p})^{-1} q(\mathbf{p})$$

- While tempting, this doesn't work. (It is not a contraction).
- There is a modification that is a contraction for logit type models.
- You can always get lucky(!)

#3: Solving Pricing Equilibria: Morrow and Skerlos (2011)

- For the logit (and variants) we can factor $\frac{\partial q_j}{\partial p_k}$ into two parts.

$$\Omega_{jk}(\mathbf{p}) = \underbrace{\alpha \cdot I[j = k] \cdot s_j(\mathbf{p})}_{\Lambda(\mathbf{p})} - \underbrace{\alpha \cdot s_j(\mathbf{p}) s_k(\mathbf{p})}_{\Gamma(\mathbf{p})}$$

- $\Gamma(\mathbf{p})$ and $\Lambda(\mathbf{p})$ are $J \times J$ matrices and $\Lambda(\mathbf{p})$ is diagonal and (j, k) is nonzero in $\Gamma(\mathbf{p})$ only if (j, k) share an owner.
- After factoring we can rescale by $\Lambda^{-1}(\mathbf{p})$

$$(\mathbf{p} - \mathbf{mc}) \leftarrow \Lambda^{-1}(\mathbf{p}) \cdot \Gamma(\mathbf{p}) \cdot (\mathbf{p} - \mathbf{mc}) - \Lambda^{-1}(\mathbf{p}) \cdot s(\mathbf{p})$$

- This alternative fixed point is in fact a contraction.
- Moreover the rate of convergence is generally fast and stable (much more than Gauss-Seidel or Gauss-Jacobi).

Some Monte Carlos re: Instruments

Differentiation Instruments: Gandhi Houde (2016)

- Need instruments for the θ_2 random coefficient parameters.
- Instead of average of other characteristics $h(x) = \frac{1}{J-1}\sum_{k \neq j} x_k$, can transform as distance to x_j.

$$d_{jt}^k = x_k - x_j$$

- And use this transformed to construct two kinds of IV (Squared distance, and count of local competitors)

$$\begin{aligned} DI\sigma_1 &= \quad \textstyle\sum_{j \in F} d_{jt}^2, \qquad \sum_{j \notin F} d_{jt}^2 \\ DI\sigma_2 &= \quad \textstyle\sum_{j \in F} I[d_{jt} < c] \qquad \sum_{j \notin F} I[d_{jt} < c] \end{aligned}$$

- They choose c to correspond to one standard deviation of x across markets.

Optimal Instruments: Reynaert Verboven (2014)

Table 2: Bias and Efficiency with Imperfect Competition

		Single Equation GMM								
		g_{jt}^1			g_{jt}^2			g_{jt}^3		
	True	Bias	St Err	RMSE	Bias	St Err	RMSE	Bias	St Err	RMSE
β^0	2	-0.127	0.899	0.907	-0.155	0.799	0.814	-0.070	0.514	0.519
β^1	2	-0.068	0.899	0.901	0.089	0.766	0.770	-0.001	0.398	0.398
α	-2	0.006	0.052	0.052	0.010	0.049	0.050	0.010	0.043	0.044
σ^1	1	-0.162	0.634	0.654	-0.147	0.537	0.556	-0.016	0.229	0.229
		Joint Equation GMM								
		g_{jt}^1			g_{jt}^2			g_{jt}^3		
	True	Bias	St Err	RMSE	Bias	St Err	RMSE	Bias	St Err	RMSE
β^0	2	-0.095	0.714	0.720	-0.103	0.677	0.685	0.005	0.459	0.459
β^1	2	0.089	0.669	0.675	0.098	0.621	0.628	-0.009	0.312	0.312
α	-2	0.001	0.047	0.047	0.002	0.046	0.046	-0.001	0.043	0.043
σ^1	1	-0.116	0.462	0.476	-0.110	0.418	0.432	0.003	0.133	0.133

Bias, standard errors (St Err) and root mean squared errors (RMSE) are computed from 1000 Monte Carlo replications. Estimates are based on the MPEC algorithm and Sparse Grid integration. The instruments g_{jt}^1, g_{jt}^2.and g_{jt}^3 are defined in section 2.4 and 2.5.

					True Value				Median Bias				Median Absolute Error			
Simulation	Supply	Z_D	Z_S	Seconds	α	σ_x	σ_p	ρ	α	σ_x	σ_p	ρ	α	σ_x	σ_p	ρ
Simple	No	$[X_1, w]$	X_3	0.1	-1	3			-0.854	-0.710			0.926	0.943		
Simple	No	$[\text{BLP}, w, X_1]$	X_3	0.8	-1	3			0.111	0.022			0.227	0.418		
Simple	No	$[\text{Local}, w, X_1]$	X_3	0.4	-1	3			0.079	0.021			0.241	0.288		
Simple	No	$[\text{Quadratic}, w, X_1]$	X_3	0.5	-1	3			0.090	-0.026			0.250	0.373		
Simple	No	$[\text{Optimal}, X_1]$	$[\text{Optimal}, X_3]$	1.1	-1	3			0.157	-0.028			0.246	0.181		
Simple	Yes	$[X_1, w]$	X_3	0.8	-1	3			-0.216	-0.410			0.337	0.814		
Simple	Yes	$[\text{BLP}, w, X_1]$	X_3	2.2	-1	3			0.111	0.022			0.226	0.418		
Simple	Yes	$[\text{Local}, w, X_1]$	X_3	1.1	-1	3			0.077	0.021			0.240	0.287		
Simple	Yes	$[\text{Quadratic}, w, X_1]$	X_3	1.2	-1	3			0.090	-0.026			0.251	0.373		
Simple	Yes	$[\text{Optimal}, X_1]$	$[\text{Optimal}, X_3]$	4.3	-1	3			0.010	0.011			0.171	0.159		
Complex	No	$[X_1, w]$	X_3	0.2	-1	3	0.2		-0.835	-0.657	-0.027		0.913	0.965	0.062	
Complex	No	$[\text{BLP}, w, X_1]$	X_3	2.8	-1	3	0.2		-0.014	-0.453	-0.016		0.269	0.826	0.200	
Complex	No	$[\text{Local}, w, X_1]$	X_3	1.3	-1	3	0.2		0.010	-0.185	-0.109		0.365	0.364	0.200	
Complex	No	$[\text{Quadratic}, w, X_1]$	X_3	1.2	-1	3	0.2		0.104	0.032	-0.200		0.353	0.403	0.200	
Complex	No	$[\text{Optimal}, X_1]$	$[\text{Optimal}, X_3]$	3.8	-1	3	0.2		-0.033	-0.130	0.089		0.272	0.248	0.174	
Complex	Yes	$[X_1, w]$	X_3	1.1	-1	3	0.2		-0.217	-0.352	-0.009		0.349	0.773	0.051	
Complex	Yes	$[\text{BLP}, w, X_1]$	X_3	7.5	-1	3	0.2		-0.030	-0.487	0.032		0.265	0.854	0.200	
Complex	Yes	$[\text{Local}, w, X_1]$	X_3	3.9	-1	3	0.2		-0.001	-0.185	-0.060		0.365	0.363	0.200	
Complex	Yes	$[\text{Quadratic}, w, X_1]$	X_3	3.5	-1	3	0.2		0.096	0.038	-0.200		0.354	0.412	0.200	
Complex	Yes	$[\text{Optimal}, X_1]$	$[\text{Optimal}, X_3]$	11.3	-1	3	0.2		-0.063	-0.041	0.044		0.185	0.179	0.136	
RCNL	No	$[X_1, w]$	X_3	0.4	-1	3		0.5	-0.846	-0.576		0.099	0.992	0.907		0.144
RCNL	No	$[\text{BLP}, w, X_1]$	X_3	5.6	-1	3		0.5	0.139	-0.717		0.028	0.274	1.168		0.063
RCNL	No	$[\text{Local}, w, X_1]$	X_3	3.0	-1	3		0.5	0.126	-0.106		0.009	0.241	0.340		0.047
RCNL	No	$[\text{Quadratic}, w, X_1]$	X_3	3.2	-1	3		0.5	0.142	-0.117		0.011	0.240	0.407		0.047
RCNL	No	$[\text{Optimal}, X_1]$	$[\text{Optimal}, X_3]$	7.6	-1	3		0.5	0.111	-0.012		-0.006	0.221	0.198		0.025
RCNL	Yes	$[X_1, w]$	X_3	1.9	-1	3		0.5	-0.233	-0.279		0.026	0.321	0.749		0.119
RCNL	Yes	$[\text{BLP}, w, X_1]$	X_3	11.3	-1	3		0.5	0.139	-0.705		0.028	0.272	1.168		0.063
RCNL	Yes	$[\text{Local}, w, X_1]$	X_3	6.0	-1	3		0.5	0.126	-0.104		0.009	0.240	0.339		0.047
RCNL	Yes	$[\text{Quadratic}, w, X_1]$	X_3	6.5	-1	3		0.5	0.141	-0.116		0.011	0.240	0.407		0.047
RCNL	Yes	$[\text{Optimal}, X_1]$	$[\text{Optimal}, X_3]$	20.5	-1	3		0.5	0.008	-0.013		0.001	0.108	0.171		0.018

But form of Optimal Instruments Doesn't Matter

				True Value				Median Bias				Median Absolute Error			
Simulation	Supply	Optimality	Seconds	α	σ_x	σ_p	ρ	α	σ_x	σ_p	ρ	α	σ_x	σ_p	ρ
Simple	No	Approximate	1.1	-1	3			0.157	-0.028			0.246	0.181		
Simple	No	Asymptotic	4.2	-1	3			0.157	-0.024			0.246	0.181		
Simple	No	Empirical	4.2	-1	3			0.157	-0.020			0.245	0.180		
Simple	Yes	Approximate	4.3	-1	3			0.010	0.011			0.171	0.159		
Simple	Yes	Asymptotic	20.0	-1	3			0.029	0.014			0.180	0.154		
Simple	Yes	Empirical	20.1	-1	3			0.012	0.009			0.175	0.159		
Complex	No	Approximate	3.8	-1	3	0.2		-0.033	-0.130	0.089		0.272	0.248	0.174	
Complex	No	Asymptotic	7.8	-1	3	0.2		-0.025	-0.136	0.088		0.268	0.247	0.175	
Complex	No	Empirical	7.8	-1	3	0.2		-0.038	-0.133	0.089		0.266	0.255	0.177	
Complex	Yes	Approximate	11.3	-1	3	0.2		-0.063	-0.041	0.044		0.185	0.179	0.136	
Complex	Yes	Asymptotic	37.9	-1	3	0.2		-0.039	-0.041	0.015		0.229	0.224	0.131	
Complex	Yes	Empirical	37.4	-1	3	0.2		-0.045	-0.045	0.025		0.222	0.211	0.122	
RCNL	No	Approximate	7.6	-1	3		0.5	0.111	-0.012		-0.006	0.221	0.198		0.025
RCNL	No	Asymptotic	12.1	-1	3		0.5	0.108	-0.015		-0.006	0.222	0.195		0.024
RCNL	No	Empirical	12.1	-1	3		0.5	0.104	-0.010		-0.007	0.221	0.195		0.025
RCNL	Yes	Approximate	20.5	-1	3		0.5	0.008	-0.013		0.001	0.108	0.171		0.018
RCNL	Yes	Asymptotic	59.6	-1	3		0.5	0.012	-0.007		0.001	0.113	0.170		0.019
RCNL	Yes	Empirical	58.9	-1	3		0.5	0.014	-0.005		0.001	0.111	0.174		0.018

Does the Supply Side Help?

- BLP95 "folk lore": not identified without it.
- Armstrong: you probably need strong cost shifters w_{jt}
- RV2016: under optimal instruments suggests it isn't important.
- Most researchers leave it out.
- Supply matters when cost shifters are weak!

Simulation	γ_2	Corr(p, w)	Supply	Seconds	True Value				Median Bias				Median Absolute Error			
					α	σ_x	σ_p	ρ	α	σ_x	σ_p	ρ	α	σ_x	σ_p	ρ
Simple	0.0	0.001	No	1.1	-1	3			0.360	-0.081			0.423	0.204		
Simple	0.0	0.001	Yes	4.8	-1	3			0.010	0.006			0.211	0.165		
Simple	0.1	0.052	No	1.1	-1	3			0.266	-0.050			0.350	0.190		
Simple	0.1	0.052	Yes	4.5	-1	3			0.010	0.006			0.206	0.162		
Simple	0.2	0.102	No	1.1	-1	3			0.157	-0.028			0.246	0.181		
Simple	0.2	0.102	Yes	4.3	-1	3			0.010	0.011			0.171	0.159		
Simple	0.4	0.199	No	1.1	-1	3			0.051	-0.001			0.126	0.170		
Simple	0.4	0.199	Yes	4.2	-1	3			0.006	0.017			0.111	0.146		
Simple	0.8	0.376	No	1.1	-1	3			0.014	0.009			0.062	0.167		
Simple	0.8	0.376	Yes	4.3	-1	3			-0.000	0.009			0.060	0.143		
Complex	0.0	0.002	No	3.8	-1	3	0.2		0.107	-0.197	0.100		0.379	0.297	0.188	
Complex	0.0	0.002	Yes	12.1	-1	3	0.2		-0.073	-0.047	0.033		0.228	0.194	0.149	
Complex	0.1	0.054	No	3.7	-1	3	0.2		0.065	-0.160	0.092		0.331	0.263	0.182	
Complex	0.1	0.054	Yes	11.5	-1	3	0.2		-0.059	-0.043	0.044		0.210	0.190	0.140	
Complex	0.2	0.104	No	3.8	-1	3	0.2		-0.033	-0.130	0.089		0.272	0.248	0.174	
Complex	0.2	0.104	Yes	11.3	-1	3	0.2		-0.063	-0.041	0.044		0.185	0.179	0.136	
Complex	0.4	0.204	No	3.7	-1	3	0.2		-0.076	-0.097	0.077		0.191	0.230	0.170	
Complex	0.4	0.204	Yes	11.3	-1	3	0.2		-0.061	-0.038	0.053		0.155	0.172	0.129	
Complex	0.8	0.384	No	3.7	-1	3	0.2		-0.089	-0.091	0.067		0.157	0.221	0.167	
Complex	0.8	0.384	Yes	12.5	-1	3	0.2		-0.057	-0.038	0.048		0.129	0.163	0.125	
RCNL	0.0	0.001	No	7.5	-1	3		0.5	0.263	-0.037		-0.013	0.355	0.204		0.028
RCNL	0.0	0.001	Yes	21.6	-1	3		0.5	0.019	-0.009		0.002	0.123	0.179		0.019
RCNL	0.1	0.050	No	7.6	-1	3		0.5	0.203	-0.017		-0.010	0.309	0.196		0.027
RCNL	0.1	0.050	Yes	20.3	-1	3		0.5	0.008	-0.010		0.001	0.119	0.170		0.019
RCNL	0.2	0.097	No	7.6	-1	3		0.5	0.111	-0.012		-0.006	0.221	0.198		0.025
RCNL	0.2	0.097	Yes	20.5	-1	3		0.5	0.008	-0.013		0.001	0.108	0.171		0.018
RCNL	0.4	0.189	No	7.8	-1	3		0.5	0.041	-0.012		-0.002	0.130	0.187		0.022
RCNL	0.4	0.189	Yes	19.8	-1	3		0.5	0.008	-0.006		0.001	0.089	0.171		0.019
RCNL	0.8	0.358	No	7.8	-1	3		0.5	0.009	0.005		-0.000	0.078	0.180		0.021
RCNL	0.8	0.358	Yes	19.7	-1	3		0.5	0.005	-0.017		0.002	0.063	0.169		0.017