A Large Scale Study of the Small Sample Performance of Random Coefficient Models of Demand

Benjamin S. Skrainka

University of Chicago
The Harris School of Public Policy
skrainka@uchicago.edu

May 17, 2012

Introduction

Objectives

This talk's objectives:

- Discuss Monte Carlo experiments to characterize properties of Berry, Levinsohn, and Pakes (1995) (BLP) estimator:
  - 'BLP' characteristics IV vs. cost shifter IV
  - Asymptotics as $J \to \infty$ and $T \to \infty$
  - Finite sample bias
  - Bias of different quadrature methods
- Demonstrate power of modern software engineering tools to answer practical econometric questions, such as behavior of an estimator:
  - PADS cluster + parameter sweep
  - C++ and Eigen for implementing high performance code
  - State of the art BLP implementation
- Generate data from structural model

Overview

Estimation Infrastructure

Overview of Infrastructure

This project depends heavily on modern software engineering and numerical methods:

- Robust and speedy implementation of BLP estimation code
- Robust and speedy implementation of code to generate data
- PADS Cluster
- Data analysis scripts (R, Python, BASH)

State of the Art BLP Implementation

This study develops what will hopefully become the new reference implementation:

- Best optimization strategy: MPEC (Su & Judd, 2011)
- Best quadrature rules: SGI (Skrainka & Judd, 2011)
- Modern solver: SNOPT (Gill, Murray, & Saunders, 2002)
- Numerically robust:
  - C++
  - Eigen, a cutting edge template library for linear algebra – at least as fast as Intel MKL!
  - Higher precision arithmetic (`long double`)
  - Analytic derivatives

Rigorous Verification

Your work lacks credibility unless you verify your code:

- Unit test every function
- Write functions which are easy to verify:
  - Do one thing only, but do it well
  - Easier to understand
  - Easier to compose
- Check all derivatives against finite difference with a suitable tolerance (usually $1.0e-6$ is reasonably good)
- May require writing extra functions like profit function to verify FOC for Bertrand-Nash price equilibrium
- If you don't test it – no matter how simple – it is probably wrong!
- In addition, compared results to Dubé, Fox, & Su MATLAB code.

Finding a Global Optimum

Even with MPEC, BLP is a difficult problem to solve reliably:

- Often very flat – perhaps even non-convex!
- Used 50 starts per replication:
  - Some did not converge, especially for larger T and J
  - Some did not satisfy feasibility conditions, especially for larger T and J, despite generating initial guesses which satisfied constraints
- Restarted every successful start to make sure it converged to the same point
- Performed for both BLP and cost shifter IV

PADS Cluster

PADS cluster provides High Throughput Computing (HTC):

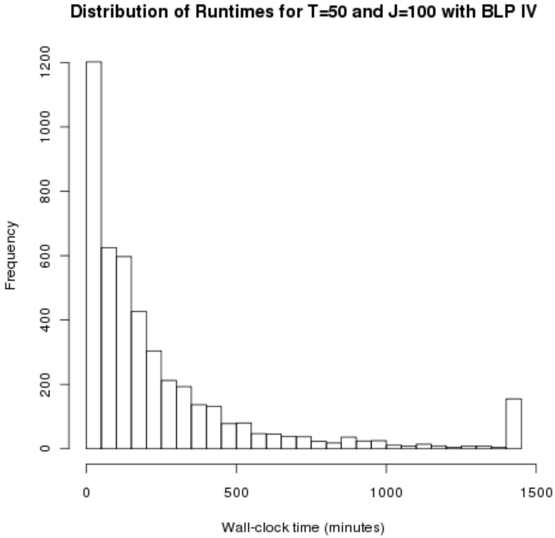
- PBS Job Manager facilitates parameter sweeps, an easy technique for parallelizing work which is independent
- Uses scripts to generate data or estimate code for some chunk of runs (1 to 50) per task
- Chunk jobs together for shorter jobs to spread scheduler overhead across more jobs
- Could never estimate BLP $> 300,000$ times on my laptop!

Parallelization

Parameter Sweep provides easy parallelization:

- Each job:
  - Estimates one or more synthetic dataset and starting value(s)
  - Short runs are chunked to minimize scheduler overhead
  - Independent of all other jobs
  - Writes results to several output files
- A separate program computes bias, RMSE, and other statistics from the output files

Job Times



Distribution of Runtimes for T=50 and J=100 with BLP IV
Frequency
0
200
400
600
800
1000
1200
0
500
1000
1500
Wall-clock time (minutes)

Dealing with Problems

Common problems running on a cluster include:

- Jobs which do not complete on time:
  - If several jobs are chunked together, must rerun with a smaller chunk size
  - Common because most starts are quick but there is a long thin tail of slow starts
- Jobs which crash:
  - Running on a large scale exposes flaws in your code
  - Must find bug in code and fix!
- Node failures and cluster crashes $\Rightarrow$ must figure out what to rerun
- Impossible to read $1,000,000$ output and error files!

Diagnosing Problems

When you scale an experiment up, you encounter many unexpected problems:

- Must automate verification as much as possible
- Perform some random inspections
- Wrote scripts to perform common tasks:
  - Compare optimum from original start and restart run
  - Check residuals when computing price equilibrium
  - Check solver exit codes
  - Compute statistics about job time used for an experiment
  - Generate PBS files
- Unix makes it easy to create tools with bash, Python, grep, sed, awk, R, etc.

Data Generation

Data Generation

Data must be generated from a structural model:

- Armstrong (2011):
  - Proves general result that for logit, nested logit, random coefficients, BLP, etc., these models are only identified as $J \to \infty$ with cost shifters.
  - I.e., BLP is unidentified with BLP instruments in large markets!
  - Corrects Berry, Linton, Pakes (2004)
  - Shows that you must generate data from a structural model or the data will not behave correctly asymptotically
- Note: each firm must produce at least two products to use BLP instruments

Intuition

Intuition comes from logit:

- FOC: $0 = s_j + (p_j - c_j)\dfrac{\partial s_j}{\partial p_j}$ or $p_j = c_j - \dfrac{s_j}{\partial s_j / \partial p_j}$
- This simplifies to: $p_j = c_j + \dfrac{1}{\alpha_{price}(1 - s_j)}$
- As $J \to \infty$, $s_j \to 0$ so product characteristics drop out of pricing equation!

Implementation

Generating synthetic data correctly is more difficult than estimating BLP!

- Must generate from a structural model (Armstrong, 2011)
- Used same software technologies (C++, Eigen, higher precision arithmetic, C++ Standard Library) as estimation code
- Used PATH to solve for price equilibrium

Solving for Price Equilibrium

Must solve for Bertrand-Nash Price Equilibrium:

- Used PATH (Ferris, Kanzow, & Munson, 1999) solver
  - Hard for large J because dense
  - Hard to solve because FOCs are highly non-linear
  - Gaussian root finding is $O\left(N^3\right)$ so becomes slow quickly as problem grows
  - BLP demand is often highly non-linear
- Solved for each market individually, exploiting block diagonal structure
- Used multiple starts per market
- Transformed problem by dividing by market shares:
  - Facilitates convergence
  - Solved original problem when transform failed

Experiments & Results

Experiments

The study performs the following experiments:

- Asymptotics
- Finite sample bias
- Bias of different quadrature methods

Design

Experiments consist of:

- The same fixed DGP (β, Σ)
- $T = \{1, 10, 25, 50\}$
- $J = \{12, 24, 48, 100\}$
- 100 replications (data sets) per experiment
- Point estimates computed using 50 starts and a restart on each optimum
- Two instrumentation strategies (BLP, Cost)
- Smaller data sets nested in larger ones
- An estimation ranges from seconds to more than 24 hours

Computational Cost

Some statistics about these experiments:

- $> 85,656$ CPU-hours
- $> 27,969$ jobs
- 16 experiments $\times$ 100 replications $\times$ 50 starts $\times$ 2 restarts $\times$ 2 IV types $= 320,000$ estimations of BLP!

Results: Overview

Bottom line: there is pronounced and persistent finite sample bias:

- Traditional BLP instruments:
  - Biased point estimates and elasticities
  - Bias always in one direction!
  - T and J not yet large enough for asymptotics to work
- Cost shifter instruments: better than BLP instruments but finite sample bias still present for most parameters
- Numerical problems increase with T and J
- pMC is more biased than SGI quadrature
- Fundamental problem: 'a few, weak instruments'

Results: $\widehat{\theta_{13}}$ – BLP IV

T	J	Bias	Mean Abs Dev	RMSE	$!CI^{95}$
1	12	−2	3	5.7	0
1	24	−0.72	1.9	3.2	0
1	48	−0.52	1.9	3	0
1	100	−0.57	1.7	2.3	0
10	12	−1.7	2.6	6	1
10	24	−0.65	2	3.6	0
10	48	−0.64	1.9	3.2	0
10	100	−0.83	2	3.9	0
25	12	−0.62	1.9	3.1	3
25	24	−0.96	2.3	3.7	1
25	48	−1.3	2.8	7.6	0
25	100	−0.95	2.1	3.7	0
50	12	−0.39	1.6	2.7	1
50	24	−1.2	2.5	5.4	1
50	48	−1.2	2.2	5.2	0
50	100	−0.63	1.9	3	0

Results: $\widehat{\theta_{13}}$ – Cost IV

T	J	Bias	Mean Abs Dev	RMSE	!CI^{95}
1	12	−0.38	1.1	1.5	1
1	24	−0.05	1	1.3	0
1	48	0.012	0.99	1.2	2
1	100	0.057	0.72	0.88	0
10	12	−0.62	1.3	2	0
10	24	−0.18	0.8	1.3	0
10	48	−0.15	0.62	0.86	0
10	100	−0.027	0.39	0.52	1
25	12	−0.38	1	1.6	0
25	24	−0.3	0.73	0.98	0
25	48	−0.11	0.45	0.63	0
25	100	−0.033	0.25	0.33	0
50	12	−0.081	0.79	1.1	0
50	24	−0.22	0.55	1	0
50	48	−0.026	0.28	0.4	0
50	100	0.003	0.19	0.26	0

Results: $\widehat{\theta_{21}}$ – BLP IV

T	J	Bias	Mean Abs Dev	RMSE	!CI^{95}
1	12	3.1	3.9	7.3	0
1	24	4.8	5.3	10	0
1	48	5.7	6.5	23	0
1	100	2.1	2.7	5.2	0
10	12	3.5	4.1	8.1	0
10	24	2.9	3.3	7.1	1
10	48	4.7	5.1	9.9	0
10	100	1.7	2.2	6.7	0
25	12	3.6	4.1	7	0
25	24	3.3	3.6	7.2	0
25	48	2.9	3.3	7.4	0
25	100	2.2	2.7	6.7	0
50	12	2.5	3	5.6	0
50	24	4.1	4.5	11	0
50	48	1.5	2	3.6	0
50	100	2.7	3.1	7.4	0

Results: $\widehat{\theta_{21}}$ – Cost IV

T	J	Bias	Mean Abs Dev	RMSE	$!CI^{95}$
1	12	7.4	8.2	13	0
1	24	8.4	8.8	14	0
1	48	7.2	8.1	13	0
1	100	6.2	7.1	12	1
10	12	0.8	1.8	2.7	0
10	24	4	4.9	11	1
10	48	2.9	3.8	6.6	0
10	100	5.9	6.8	11	0
25	12	1.5	2.3	3.4	0
25	24	3.6	4.4	7.7	0
25	48	3.7	4.6	7	1
25	100	6.2	7	11	0
50	12	0.97	2	3.1	0
50	24	3.9	4.6	12	0
50	48	3.6	4.2	6.3	1
50	100	5.9	6.6	12	0

Results: Elasticities – BLP IV

T	J	Bias	Mean Abs Dev	Med Abs Dev	RMSE
1	12	−0.77	2.2	0.94	4.9
1	24	−0.095	1.5	0.77	3.3
1	48	−0.082	1.6	0.91	2.7
1	100	−0.39	1.5	0.98	2.5
10	12	−0.5	1.7	0.81	3.3
10	24	−0.57	1.7	0.83	3.3
10	48	−0.16	1.5	0.97	2.2
10	100	−0.53	1.7	0.93	3.3
25	12	−0.3	1.4	0.94	2.7
25	24	−0.72	1.8	1.1	3
25	48	−0.87	2.2	1.1	4.9
25	100	−0.61	1.7	0.97	2.7
50	12	−0.43	1.5	0.94	2.6
50	24	−0.77	1.9	0.91	3.8
50	48	−0.97	1.9	1.1	4
50	100	−0.56	1.8	1.1	2.9

Results: Elasticities – Cost IV

T	J	Bias	Mean Abs Dev	Med Abs Dev	RMSE
1	12	0.059	0.86	0.52	1.4
1	24	0.17	0.83	0.55	1.3
1	48	0.11	0.85	0.6	1.3
1	100	−0.59	1.3	0.43	60
10	12	−0.098	0.69	0.48	1
10	24	−0.095	0.52	0.33	0.82
10	48	−0.15	0.48	0.28	4.2
10	100	−0.072	0.3	0.19	0.54
25	12	−0.23	0.56	0.38	0.83
25	24	−0.22	0.48	0.34	0.69
25	48	−0.062	0.3	0.19	0.45
25	100	−0.16	0.3	0.13	0.68
50	12	−0.27	0.54	0.32	0.92
50	24	−0.32	0.46	0.22	1
50	48	−0.1	0.2	0.12	0.33
50	100	−0.15	0.24	0.098	0.57

Results: Solver Convergence

SNOPT has increasing difficulty finding an optimum as the number of markets and products increase:

- Most common problem: cannot find a feasible point
- Other problems:
  - Hits iteration limit
  - Not enough real storage
  - Singular basis

Results: pMC vs SGI

	Bias		Mean Abs Dev		RMSE	
	SGI	pMC	SGI	pMC	SGI	pMC
θ_{11}	0.96	12.34	2.29	13.25	4.00	28.92
θ_{12}	0.02	-0.13	0.52	0.38	0.94	0.48
θ_{13}	-0.28	-0.38	1.47	1.21	3.01	1.51
θ_{21}	22.57	128.22	23.01	128.24	81.76	253.87
θ_{22}	0.02	-0.04	0.12	0.16	0.19	0.20
θ_{23}	0.08	0.64	0.36	0.75	0.75	0.90

Table: Comparison of bias in point estimates : SGI vs. pMC for T=2 markets and J=24 products with 165 nodes.

Note: pMC also shows BLP is biased if you compute p-values!

Next Steps

This infrastructure can be used to solve several related problems:

- Use bootstrap to compute p-values for bias
- Rerun experiments in Skrainka & Judd (2011) on a larger scale and compute bias for different rules
- Bootstrap BLP to study where asymptotic GMM standard errors are valid
- Evaluate other estimation approaches such as Empirical Likelihood (Conlon, 2010)

Conclusion

Developed infrastructure to test BLP estimator:

- Characterize estimator's bias for a range of markets and number of products
- Computed bias for BLP and Cost IV
- Demonstrated power of modern HTC + Monte Carlo experiments to answer questions where (econometric) theory has failed to produce an answer.
- Shown that these resources are easily accessible to economists