1 Mathematical Programming Approach to BLP

1.1 BLP Setup

In the BLP model person i's utility from choice j is

$$U_{ij} = \alpha \ln(y_i - p_j) + x_j \beta + \xi_{jm} + \sum_{l=1}^{L} \sigma_l x_{jl} v_{il} + \epsilon_{ij} \tag{1}$$

where ϵ_{ij} has an extreme value type I distribution. This implies that the population probability of choice j in market m is

$$P_{jm}(\alpha, \beta, \sigma) = \int \frac{\exp(\alpha \ln(y_i - p_{jm}) + x_j \beta + \xi_{jm} + \sum_{l=1}^{L} \sigma_l x_{jl} \nu_{il} + \epsilon_{ij})}{\sum_{j'=0}^{J} \exp(\alpha \ln(y_i - p_{j'm}) + x_{j'} \beta + \xi_{j'm} + \sum_{l=1}^{L} \sigma_l x_{j'l} v_{il} + \epsilon_{ij'})} dG_m(\nu, y_i) \tag{2}$$

If ξ_j were observable, we could estimate this model by MLE. ξ is not observarble, so we must do something else. If we thought ξ was independent from p, y, and x, then we could integrate it out. However, if firms know ξ, then firms will set their prices to depend on ξ. Consequently, we must instrument for at least prices. If our instruments are z, then we have the following moment condition

$$E[\xi_{jm} z_{jm}] = 0$$

Or equivalently if $\delta_{jm} = \xi_{jm} + x_j \beta$,

$$E[(\delta_{jm} - x_j \beta) z_{jm}] = 0 \tag{3}$$

Where we require δ_{jm} to rationalize the observed product shares, that is, δ_{jm} must satisfy

$$\hat{s}_{jm} = \int \frac{\exp(\alpha \ln(y_i - p_{jm}) + \delta_{jm} + \sum_{l=1}^{L} \sigma_l x_{jl} \nu_{il} + \epsilon_{ij})}{\sum_{j'=0}^{J} \exp(\alpha \ln(y_i - p_{j'm}) + \delta_{j'm} + \sum_{l=1}^{L} \sigma_l x_{j'l} v_{il} + \epsilon_{ij'})} dG_m(\nu, y_i) \tag{4}$$

BLP view (4) as an equation that implicitly defines a function mapping the other parameters to $\delta(\alpha, \sigma)$. They then propose estimating the parameters by minimizing

$$\min_{\alpha, \beta, \sigma} (\delta(\alpha, \sigma) - x\beta)' z' W z (\delta(\alpha, \sigma) - x\beta) \tag{5}$$

This is a difficult function to minimize because $\delta(\alpha, \beta)$ is difficult to compute. Computing $\delta(\alpha, \beta)$ requires solving the nonlinear system of equations (4). This general situation – where you want to minimize an objective function that depends on another, hard to compute function – is quite common in structural econometrics. It occurs anytime the equilibrium of your model does not have a closed form solution. Examples include models of games and models of investment.

1.2 Mathematical Programming

The mathematical programming approach to BLP rewrites the problem as a constrained optimization problem:

$$\min_{\alpha, \beta, \sigma, \delta} (\delta - x\beta)' z' W z (\delta - x\beta)$$

$$s.t.$$

$$\hat{s}_{jm} = \int \frac{\exp(\alpha \ln(y_i - p_{jm}) + \delta_{jm} + \sum_{l=1}^{L} \sigma_l x_{jl} \nu_{il} + \epsilon_{ij})}{\sum_{j'=0}^{J} \exp(\alpha \ln(y_i - p_{j'm}) + \delta_{j'm} + \sum_{l=1}^{L} \sigma_l x_{j'l} v_{il} + \epsilon_{ij'})} dG_m(\nu, y_i) \tag{6}$$

This probably does not look any easier to solve. However, constrained optimization problems are very common, and they have been heavily studied by computer scientists and mathematicians. Consequently, there are many good algorithms and software tools for solving constrained optimization problems. These algorithms are often faster than the plug in the constraint approach of BLP. Part of the reason is that by not always requiring the constraint to be satisfied, mathematical programming algorithms can adjust all the parameters to simultaneously lower the objective function value and make the constraints closer to being satisfied. Ken Judd and coauthors have a few papers advocating the use of mathematical programming in economics.

1.3 BLP in AMPL

AMPL (**A** **M**athematical **P**rogramming **L**anguage) is a computer language for mathematical programming. The advantages of AMPL are:

- Simple, natural syntax for writing down mathematical programs
- Automatically computes derivatives, which are needed to use the most efficient solution algorithms
- Provides a simple interface to a large number of state of the art solution algorithms

AMPL's disadvantages:

- Simple syntax and limited set of commands makes it painful to do anything other than write down mathematical programs
- Very memory intensive
- Poorly documented
- Free student version limited to 300 variables and 300 constraints

I wrote some code to simulate and estimate a BLP type model in AMPL. You can find the code on the webpage. Strangely enough, the hardest part of writing the program was figuring out to simulate the model. The problem was that with a poorly chosen DGP, it is quite likely that there is no value of the parameters that both match the simulated shares and satisfy the supply side restrictions. If you include a supply side in the estimation, then

$$mc_{jm} = p_{jm} + \frac{s_{jm}}{\partial s_{jm}/\partial p}(\alpha, \delta, \sigma)$$

It's natural to require $mc \geq 0$. Unfortunately, there is no guarantee that there is there exists parameters such that $\hat{s}_{jm} = P_{jm}(\alpha, \delta, \sigma)$ and $p_{jm} + \frac{s_{jm}}{\partial s_{jm}/\partial p}(\alpha, \delta, \sigma) \geq 0$. With enough simulations to create $\hat{s}_{jm}$ you are guaranteed that a solution exists, but with a small number of simulations, you are not. I used 100 simulations, and had to adjust the parameters and distribution of covariates to avoid problems. [1]

The model I estimated has an extra set of parameters. Utility is given by

$$U_{ij} = \alpha \ln(y_i - p_{jm}) + x_j\beta + \xi_{jm} + \sum_{l=1}^{L} \sigma_l x_{jl} v_{il} + \sum_{k=1}^{K}\sum_{l=1}^{L} \pi_{kl} x_{jl} d_{ik} + \epsilon_{ij} \tag{7}$$

where d_{ik} are observed demographic characteristics. For each market, d_{ik} were drawn with replacement from the empirical distribution of the characteristics. The idea here is that our markets are something like states or years, and we can look at the census to find out the distribution of demographics in each market.

I also estimated the model with supply side moments. As in Whitney's notes, I assume Bertrand competition and marginal cost equal to

$$mc_{jm} = \exp(c_j\alpha_c + \eta_{jm}) \tag{8}$$

The first order condition for firms is:

$$s_{jm}(p, x, \xi; \theta) + (p_{jm} - mc_{jm})\frac{\partial s_{jm}}{\partial p} = 0 \tag{9}$$

I have assumed that each firm only produces one product, so that this equation is slightly simpler than in the lecture notes. Rearranging the first order condition gives:

$$\eta_{jm} = \ln\left(p + \frac{s_{jm}}{(\partial s_{jm})/(\partial p)}\right) - c_j\alpha_c \tag{10}$$

[1] This experience made me wonder: why not treat $\hat{s}_{jm} = P_{jm}(\cdot)$ as moments instead of constraints? I suppose the answer is that $\hat{s}_{jm}$ is often computed from population data, and so is without error. However, there must be some applications where $\hat{s}_{jm}$ comes from a smaller sample, and then I think treating everything as a moment would make more sense.

If we have some instruments correlated with the right hand side of the equation, but uncorrelated with η_{jm}, our moments are:

$$0 = E\left[z_c\left(\ln\left(p + \frac{s_{jm}}{(\partial s_{jm})/(\partial p)}\right) - c_j\alpha_c\right)\right] \tag{11}$$

1.3.1 Model File

```
# AMPL model file for BLP demand estimation with supply side
# Paul Schrimpf
# 5 November, 2008

# We assume Bertrand competition with each product produced
# by a different firm.

# data
param Kp; # product characteristics
param Ki; # individual characterisitcs
param J;  # number of products
param N;  # number of individuals (only observe their characteristics)
param S;  # number of draws of random coefs (and from observed char dist)
param M;  # number of markets
param Kz; # number of demand instruments
param Kc; # number of cost shifters
param Kzc; # number of cost instruments
param c{1..J,1..M,1..Kc}; # cost shitters (notation slightly diff than in notes)
param x{1..J,1..Kp};
param w{1..N,1..M,1..Ki}; # individual characteristics (notation slightly diff than in notes)
param yo{1..N,1..M}; # observed income distribution
param z{1..J,1..M,1..Kz}; # demand instruments
param zc{1..J,1..M,1..Kzc}; # supply instruments
param s{1..J,1..M}; # product shares
param W{1..Kz,1..Kz,1..M,1..M}; # demand weighting matrix
param Wc{1..Kzc,1..Kzc,1..M,1..M}; # supply weighting matrix
 # Note: assuming block diagonal weighting between demand and supply
param nu{1..S,1..M,1..Kp}; # draws for random coefs
param d{1..S,1..M,1..Ki};  # draws for individual characteristics
param y{1..S,1..M}; # simulated income distribution

var p{1..J,1..M} >= 0; # prices
 # if we had real data, prices would be a parameter, but we're simulating
 # so we need to solve for prices, so they're a variable
param etaSim{1..J,1..M};

# variables (things that will be solved for)
var delta{1..J,1..M};
var alpha; # price coef
var beta{1..Kp}; # mean coefs on x
var pi{1..Kp,1..Ki}; # x*w coefs
var sig{1..Kp}>=0; # std dev of coefs on x0
var alphac{1..Kc}; # coefs on c

# parts of computation
var omega{j in 1..J,m in 1..M} # aka xi in notes
  = (delta[j,m] - sum{k in 1..Kp} x[j,k]*beta[k]);
var mu{j in 1..J,m in 1..M, i in 1..S} =
   delta[j,m] + alpha*log(max(1e-307,y[i,m] - p[j,m])) + sum{k in 1..Kp} x[j,k]*
   (sig[k]*nu[i,m,k] + sum{ki in 1..Ki} d[i,m,ki]*pi[k,ki]);
var pbuy{i in 1..S, j in 1..J,m in 1..M} = # P(sim i buys product j)
     exp(mu[j,m,i])
   / (exp(alpha)*y[i,m] + sum{j2 in 1..J} exp(mu[j2,m,i]));
var shat{j in 1..J, m in 1..M} = # share implied by model
       1/S*sum{i in 1..S} pbuy[i,j,m];
var dsdp{j in 1..J,m in 1..M} = # derivative of share wrt price
  1/S * sum{i in 1..S} -alpha/max(1e-307,y[i,m]-p[j,m])*pbuy[i,j,m]*(1-pbuy[i,j,m]);
var eta{j in 1..J,m in 1..M} = # shocks to marginal cost
  log(max(1e-307,p[j,m] + shat[j,m]/dsdp[j,m]))  # avoid log(0)
  - sum{k in 1..Kc} alphac[k]*c[j,m,k];
```

```

minimize moments :
  sum{m1 in 1..M,kz1 in 1..Kz,m2 in 1..M,kz2 in 1..Kz}
  (sum{j in 1..J} omega[j,m1]*z[j,m1,kz1])
  *W[kz1,kz2,m1,m2]*
  (sum{j in 1..J} omega[j,m2]*z[j,m2,kz2])
  +
  sum{m1 in 1..M,kz1 in 1..Kzc,m2 in 1..M,kz2 in 1..Kzc}
  (sum{j in 1..J} eta[j,m1]*zc[j,m1,kz1])
  *Wc[kz1,kz2,m1,m2]*
  (sum{j in 1..J} eta[j,m2]*zc[j,m2,kz2])
  ;

subject to shares{j in 1..J,m in 1..M}:
  s[j,m] = shat[j,m];

subject to niceP{j in 1..J, m in 1..M}:
  p[j,m] + shat[j,m]/dsdp[j,m] >= 0;
  # if not, it's impossible to satisfy firm FOC, and eta will be NaN

subject to priceCon{j in 1..J,m in 1..M}:
  p[j,m] = exp((sum{k in 1..Kc} alphac[k]*c[j,m,k])+etaSim[j,m])
    - shat[j,m]/dsdp[j,m];
```

1.3.2 Command File

```
# set up data
model blpSupply.mod; # tells ampl which .mod file to use

# set size of data
let Kp := 1; # number of product characteristics
let Ki := 0; # number of individual characteristics
let J := 3; # number of products
let N := 100; # number of individual observations
let M := 50; # number of markets / time periods
let Kz := 4; # number of instruments
let Kc := 1; # number of cost shifters (first one is a constant)
let Kzc := 2*Kc+1; # number of cost instruments

option randseed 270;

# write header in output file
printf 'price_solve_num,solve_result_num (0=success),' >> blpD.csv;
printf 'alpha,' >> blpD.csv;
printf{k in 1..Kp} 'beta %d,', k >> blpD.csv;
printf{k in 1..Kp} 'sig %d,', k >> blpD.csv;
printf{k in 1..Kp,ki in 1..Ki} 'pi %d %d,', k,ki >> blpD.csv;
printf{k in 1..Kc} 'alphac %d,', k >> blpD.csv;
printf{j in 1..J,m in 1..M} 'delta %d %d,',j,m >> blpD.csv;
printf '\n' >> blpD.csv;

# variables used in simulation
param dim{i in 1..S,m in 1..M};
param dm{i in 1..S,m in 1..M};
param xi{j in 1..J,m in 1..M};
param eps{j in 1..J,m in 1..M,i in 1..S};
param choice{1..M,1..S};
param maxU;
param pSolve;

for{mc in 1..100} { # do many repetitions
  # reset parameters
  let{k in 1..Kp} beta[k] := 1;
  let{k in 1..Kp, ki in 1..Ki} pi[k,ki] := 1;
  let{k in 1..Kp} sig[k] := 1;
  let{k in 1..Kc} alphac[k] := 1;
  let alpha := 1;

```

```
  let{kz1 in 1..Kz,kz2 in 1..Kz,m1 in 1..M,m2 in 1..M}
    W[kz1,kz2,m1,m2] := if (kz1=kz2 and m1==m2) then 1 else 0;
  # don't want to use supply moments, so set Wc to 0.
  let{kz1 in 1..Kzc,kz2 in 1..Kzc,m1 in 1..M,m2 in 1..M}
    Wc[kz1,kz2,m1,m2] := if (kz1=kz2 and m1==m2) then 1 else 0; #0;
  let{j in 1..J,m in 1..M} s[j,m] := 1/J;

  include simulateData.ampl;
  let pSolve := solve_result_num;

  # estimation section
  let S := 100;
  # redraw nu and d to be fair
  let{i in 1..S,m in 1..M,k in 1..Ki} d[i,m,k] := w[ceil(Uniform(0,N)),m,k];
  let{i in 1..S,m in 1..M} y[i,m] := yo[ceil(Uniform(0,N)),m];
  let{i in 1..S,m in 1..M,k in 1..Kp} nu[i,m,k] := Normal(0,1);

  # choose solver and set options
  let alpha := alpha/2;
  option solver snopt;
  option snopt_options 'iterations=10000 outlev 1 timing 1';

  fix p; drop priceCon; drop niceP;
  restore moments; restore shares;
  unfix alpha; unfix beta; unfix delta; unfix pi; unfix sig; unfix alphac;
  solve;
  #display results
#  display delta;
#  display {j in 1..J} 1/(M*S)*sum{m in 1..M,i in 1..S} mu[j,m,i];
#  display{j in 1..J} 1/(M*S)*sum{m in 1..M,i in 1..S}
#    exp(mu[j,m,i]) / (1 + sum{j2 in 1..J} exp(mu[j2,m,i]));
#  display{j in 1..J} 1/M*sum{m in 1..M} s[j,m];
  display beta,sig,pi;
  display alpha;
  #display {j in 1..J} 1/M*(sum{m in 1..M} eta[j,m]);
  #display p,eta,etaSim,omega;

  # print results
  printf '%d,%d,', pSolve,solve_result_num >> blpD.csv;
  printf '%.8g,', alpha >> blpD.csv;
  printf{k in 1..Kp} '%.8g,', beta[k] >> blpD.csv;
  printf{k in 1..Kp} '%.8g,', sig[k] >> blpD.csv;
  printf{k in 1..Kp, ki in 1..Ki} '%.8g,', pi[k,ki] >> blpD.csv;
  printf{k in 1..Kc} '%.8g,', alphac[k] >> blpD.csv;
  printf{j in 1..J,m in 1..M} '%.8g,', delta[j,m] >> blpD.csv;
  printf '\n' >> blpD.csv;
  display mc;
}
```

1.3.3 Data Simulation Comands

```
  # simulate data
  let S := N;
  let{i in 1..S, m in 1..M} y[i,m] := exp(Normal(2,1));
  let{i in 1..N, m in 1..M} yo[i,m] := y[i,m];
  let{j in 1..J,m in 1..M,i in 1..S} eps[j,m,i] :=
      -log(-log(Uniform(0,1)));
  let{j in 1..J,m in 1..M} xi[j,m] := Normal(0,1);
  display 1/(J*M*S)*sum{j in 1..J,m in 1..M,i in 1..S} eps[j,m,i];
  # instruments =
  # (constant, sum other xi, things correlated with x's but not xi)
  let{j in 1..J,m in 1..M} z[j,m,1] := 1; # include constant as instrument
  let{j in 1..J,m in 1..M} z[j,m,2] := 1/J*sum{j2 in 1..J}
         if (j2==j) then 0 else xi[j,m];
  let {j in 1..J,m in 1..M,k in 3..Kz} z[j,m,k] := Normal(0,1);
  #let {j in 1..J} x[j,1] := 1;
  let {j in 1..J,k in 1..Kp} x[j,k] := 1/sqrt(M)*(sum{m in 1..M} xi[j,m]) +
                + (sum{kz in 3..Kz,m in 1..M} z[j,m,kz])/sqrt((Kz-2)*M)
```

```
                + Normal(0,1);
  let{j in 1..J,m in 1..M} delta[j,m] := xi[j,m] + sum{k in 1..Kp} x[j,k]*beta[k];
  let{m in 1..M,k in 1..Ki} dm[m,k] := Normal(0,1);
  let{i in 1..S, m in 1..M} dim[i,m] := Normal(0,1);
  let{i in 1..S,m in 1..M,k in 1..Ki} d[i,m,k] := dm[m,k] + dim[i,m] + Normal(0,1);
  let{i in 1..N,m in 1..M,k in 1..Ki} w[i,m,k] := d[i,m,k];
  let{i in 1..S,m in 1..M,k in 1..Kp} nu[i,m,k] := Normal(0,1);

  # find prices
  let{j in 1..J,m in 1..M} c[j,m,1] := 1;
  let{j in 1..J,m in 1..M,k in 2..Kc} c[j,m,k] := Normal(0,0.1);
  let{j in 1..J,m in 1..M,k in 1..Kc} zc[j,m,k] := c[j,m,k];
  let{j in 1..J,m in 1..M,k in 1..Kc} zc[j,m,k+Kc] := c[j,m,k]^2;
  let{j in 1..J,m in 1..M} zc[j,m,2*Kc+1] := 1;
  let{j in 1..J,m in 1..M} etaSim[j,m] := Normal(0,0.1);
  let{j in 1..J,m in 1..M} p[j,m] := exp((sum{k in 1..Kc} alphac[k]*c[j,m,k])+etaSim[j,m]);
  drop moments; drop shares;
  fix alpha; fix beta; fix alphac; fix delta; fix pi; fix sig;
  unfix p; restore priceCon;
  option solver snopt;
  option snopt_options 'iterations=10000 outlev 1 timing 1';
  solve;

  # simulate choices
  for{m in 1..M,i in 1..S} {
    let maxU := alpha*log(y[i,m])-log(-log(Uniform(0,1))); # outside option
    let choice[m,i] := 0;
    for{j in 1..J} {
      if (mu[j,m,i]+eps[j,m,i] >= maxU) then {
         let maxU := mu[j,m,i]+eps[j,m,i];
         let choice[m,i] := j;
      }
    }
  }
  #let{j in 1..J,m in 1..M} s[j,m] := 1/S * sum{i in 1..S}
  #  (if (choice[m,i]=j) then 1 else 0);

  # lower simulation error by setting s = E[s]
  let{j in 1..J,m in 1..M} s[j,m] := 1/S * sum{i in 1..S} pbuy[i,j,m];

  # done simulating, let's print some summary stats
  display {j in 1..J} 1/M*(sum{m in 1..M} s[j,m]);
  display {j in 1..J} 1/M*(sum{m in 1..M} p[j,m]);
  # check that instruments are correlated with prices
  display {k in 2..Kz,j in 1..J}
     (sum{m in 1..M} (p[j,m]-(sum{m1 in 1..M} p[j,m1])/M)*
                     (z[j,m,k]-(sum{m1 in 1..M} z[j,m1,k])/M))
     / sqrt((sum{m in 1..M} (p[j,m]-(sum{m1 in 1..M} p[j,m1])/M)^2)*
            (sum{m in 1..M} (z[j,m,k]-(sum{m1 in 1..M} z[j,m1,k])/M)^2));
#  display p,eta,etaSim,omega;
#  display {j in 1..J} sum{k in 1..Kp} x[j,k]*beta[k];
#  display {j in 1..J} 1/M*(sum{m in 1..M} delta[j,m]);
#  display {j in 1..J} 1/(M*S)*sum{m in 1..M,i in 1..S} mu[j,m,i];
#  display{j in 1..J} 1/(M*S)*sum{m in 1..M,i in 1..S}
#    exp(mu[j,m,i]) / (1 + sum{j2 in 1..J} exp(mu[j2,m,i]));
```

1.3.4 Results

These results look rather dismal. In tables 1-4, the only parameters that are well-estimated (perhaps too well estimated) are the coefficients on cost shifters. These are also the only exogenous covariates in the model. In the simulations all the product characteristics are endogenous. I checked that my instrument are correlated with the x's and p's and they are. The results in tables 5 and 6, where there are fewer parameters and more markets are better, but still not all that precise. The estimates of σ are uniformly downward biased. Note that I simply used an identity weighting matrix. Perhaps optimally weighting GMM would work better. I guess it's also possible that there's a bug in my code, but the reasonably good results in tables 5 and 6 give me some confidence.

Table 1: Demand Only, 10 Markets

Parameter	True Value	Mean	Median	Variance	MSE
alpha	0.1	0.384316	0.275705	0.180023	0.258813
beta1	1	0.625504	0.74283	1.84643	1.96569
beta2	1	0.412645	0.580706	2.35841	2.67659
sig1	1	0.496225	0.280457	0.325916	0.576002
sig2	1	0.394873	0.158396	0.264492	0.627664
Pi1	1	0.419959	0.503488	0.315361	0.648225
Pi2	1	0.369032	0.327425	0.308887	0.703498
Pi3	1	0.393763	0.389155	0.301791	0.665885
Pi4	1	0.455244	0.39654	0.527863	0.818623
Based on 88 Repititions					

Table 2: Supply and Demand , 10 Markets

Parameter	True Value	Mean	Median	Variance	MSE
alpha	0.1	0.366459	0.34328	0.121739	0.191324
beta1	1	0.676725	0.613693	1.48925	1.57644
beta2	1	0.512364	0.610556	1.70191	1.91991
sig1	1	0.61693	0.325095	0.54897	0.689329
sig2	1	0.475165	0.126225	0.461227	0.731316
Pi1	1	0.463938	0.493806	0.218257	0.503082
Pi2	1	0.406713	0.389842	0.219308	0.568748
Pi3	1	0.447171	0.452772	0.273707	0.576144
Pi4	1	0.4613	0.409517	0.26424	0.551365
alphac1	1	0.999108	0.998652	0.000278784	0.000276339
alphac2	1	1.02178	1.03259	0.128093	0.127078
Based on 86 Repititions					

Table 3: Demand Only, 37 Markets

Parameter	True Value	Mean	Median	Variance	MSE
alpha	0.1	0.339158	0.276266	0.109627	0.165436
beta1	1	0.318094	0.344663	0.532033	0.990294
beta2	1	0.459523	0.433774	0.809841	1.0917
sig1	1	0.361701	0.144936	0.19566	0.600609
sig2	1	0.407519	0.218696	0.233244	0.581326
Pi1	1	0.344171	0.336979	0.0627252	0.492043
Pi2	1	0.366683	0.395075	0.0659997	0.466254
Pi3	1	0.397144	0.303257	0.138205	0.499891
Pi4	1	0.333043	0.316097	0.096952	0.540557
Based on 79 Repititions					

Table 4: Supply and Demand, 37 Markets

Parameter	True Value	Mean	Median	Variance	MSE
alpha	0.1	0.293901	0.127863	0.108168	0.144263
beta1	1	0.590004	0.63688	0.531229	0.691948
beta2	1	0.289501	0.313651	0.705877	1.20088
sig1	1	0.429779	0.140768	0.265643	0.587106
sig2	1	0.385324	0.179484	0.260532	0.63474
Pi1	1	0.385228	0.386784	0.0984804	0.475057
Pi2	1	0.415153	0.368206	0.253622	0.592146
Pi3	1	0.382904	0.364924	0.146265	0.525041
Pi4	1	0.319048	0.292805	0.266855	0.726845
alphac1	1	0.998876	0.999012	6.54546e-05	6.58082e-05
alphac2	1	0.983249	0.967823	0.0310205	0.0308702
Based on 72 Repititions					

Table 5: Demand Only, 50 Markets, fewer Parameters

Parameter	True Value	Mean	Median	Variance	MSE
alpha	1	0.806884	0.884424	0.0706302	0.107063
beta1	1	0.834611	0.838063	0.138823	0.164484
sig1	1	0.334869	0.199613	0.114434	0.555439
Based on 82 Repititions					

Table 6: Supply and Demand, 50 Markets, fewer Parameters

Parameter	True Value	Mean	Median	Variance	MSE
alpha	1	0.787458	0.892897	0.0609091	0.10526
beta1	1	0.864995	0.892974	0.139608	0.155948
sig1	1	0.337431	0.248966	0.113554	0.551018
alphac1	1	1.23982	1.24512	0.0072212	0.0646391
Based on 74 Repititions					

2 Weakly Identified GMM: Estimation of the NKPC

One popular use of GMM in applied macro has been estimating the Neo-Keynesian Phillips Curve.[2] An important example is Galí and Gertler (1999). This is an interesting paper because it involves a good amount of macroeconomics, validated a model that many macroeconomists like, and (best of all for econometricians) has become a leading example of weak identification in GMM. These notes describe Galí and Gertler's paper, then give a quick overview of identification robust inference in GMM, and finally describe the results of identification robust procedures for Galí and Gertler's models.

2.1 Deriving the NKPC

You should have seen this in macro, so I'm going to go through it quickly. Suppose there is a continuum of identical firms that sell differentiated products to a representative consumer with Dixit-Stiglitz preferences over the goods. Prices are sticky in the sense of Calvo (1983). More specifically, each period each firm has a probability of $1-\theta$ of being able to adjust its price each period. If p_t^* is the log price chosen by firms that adjust at time t, then the evolution of the log price level will be

$$p_t = \theta p_{t-1} + (1-\theta)p_t^* \tag{12}$$

The first order condition (or maybe a first order approx to the first order condition) for firms that get to adjust their price at time t is

$$p_t^* = (1-\beta\theta)\sum_{k=0}^{\infty}(\beta\theta)^k E_t[mc_{t+k}^n + \mu] \tag{13}$$

where mc_t^n is log nominal marginal cost at time t, and μ is a markup parameter that depends on consumer preferences. This first order condition can be rewritten as:

$$\begin{aligned} p_t^* =&(1-\beta\theta)mc_t^n + (1-\beta\theta)\sum_{k=1}^{\infty}(\beta\theta)^k E_t[mc_{t+k}^n + \mu] \\ =&(1-\beta\theta)(\mu + mc_t^n) + (1-\beta\theta)\beta\theta E_t p_{t+1}^* \end{aligned}$$

Substituing in $p_t^* = \frac{p_t - \theta p_{t-1}}{1-\theta}$ gives:

$$\begin{aligned} \frac{p_t - \theta p_{t-1}}{1-\theta} =& \frac{1-\beta\theta}{1-\theta}E_t[p_{t+1} - \theta p_t] + (1-\beta\theta)(\mu + mc_t^n) \\ p_t - p_{t-1} =& \beta E_t[p_{t+1} - p_t] + \frac{(1-\theta)(1-\theta\beta)}{\theta}(\mu + mc_t^n - p_t) \\ \pi_t =& \beta E_t[\pi_{t+1}] + \lambda(\mu + mc_t^n - p_t) \end{aligned} \tag{14}$$

(15)

This is the NKPC. Inflation depends on expected inflation and real marginal costs (or the deviation of log marginal costs from the steady state. In the steady state $\mu - p = mc$.).

2.2 Estimation

Using the Output Gap Since real marginal costs are difficult to observe, people have noted that in a model without capital, $mc_t^n - p_t \approx \kappa x_t$ where x_t is the output gap (the difference between current output and output in a model without price frictions). This suggests estimating:

$$\beta\pi_t = \pi_{t-1} - \lambda\kappa x_t - \lambda\mu + \epsilon_t$$

When estimating this equation, people general find that $\widehat{-\lambda\kappa}$ is positive, contradicting the model.

[2] I wrote this part of these notes for time series, so they feature a time series type application. However, the section about weak identification and identification robust inference is very relevant for this class.

GG Galí and Gertler (1999) argued that there at least two problems with this model: (i) the output gap is hard to measure and (ii) the output gap may not be proportional to real marginal costs. Galí and Gertler argue that the labor income share is a better proxy for real marginal costs. With a Cobb-Douglas production function,

$$Y_t = A_t K_t^{\alpha_k} L_t^{\alpha_l}$$

marginal cost is the ratio of the wage to the marginal product of labor,

$$\begin{aligned} MC_t = & \frac{W_t}{P_t(\partial Y_t/\partial Y)} = \frac{W_t L_t}{P_t \alpha_l Y_t} \\ = & \frac{1}{\alpha_l} S_{Lt} \end{aligned}$$

Thus the deviation of log marginal cost from its steady state should equal the deviation of log labor share from its steady state, $mc_t = s_t$. This leads to moment conditions:

$$E_t[(\pi_t - \lambda s_t - \beta\pi_{t+1})z_t] = 0 \tag{16}$$

$$E_t[(\theta\pi_t - (1-\theta)(1-\beta\theta)s_t - \theta\beta\pi_{t+1})z_t] = 0 \tag{17}$$

where z_t are any variables in firms' information sets at time t. As instruments, Galí and Gertler use four lags of inflation, the labor income share, the output gap, the long-short interest rate spread, wage inflation, and commodity price inflation. Galí and Gertler estimate this model and find values of β around 0.95, θ around 0.85, and λ around 0.05. In particular, $\lambda > 0$ in accordance with the theory unlike when using the output gap. The estimates of θ are a bit high. They imply an average price duration of five to six quarters, which is much higher than observed in the micro-data of Bils and Klenow (200?).

2.3 Hybrid Philips Curve

The NKPC implies that price setting behavior is purely forward looking. All inflation inertia comes from price stickiness in this model. One might be concerned whether this is enough to capture the observed dynamics of inflation. To answer this question, Galí and Gertler consider a more general model that allows for backward looking behavior. In particular, they assume that a fraction, ω of firms set prices equal to the optimal price last period plus an inflation adjustment: $p_t^b = p_{t-1}^* + \pi_{t-1}$. The rest of the firms behave optimally. This leads to the following inflation equation:

$$\begin{aligned} \pi_t = & \frac{(1-\omega)(1-\theta)(1-\beta\theta)mc_t + \beta\theta E_t\pi_{t+1} + \omega\pi_{t-1}}{\theta + \omega(1-\theta(1-\beta))} \\ = & \lambda mc_t + \gamma^f E_t\pi_{t+1} + \gamma^b \pi_{t-1} \end{aligned} \tag{18}$$

As above, Galí and Gertler estimate this equation using GMM. The find $\hat{\omega} \approx 0.25$ with a standard error of 0.03, so a purely forward looking model is rejected. Their estimates of θ and β are roughly the same as above.

2.4 Identification Issues

Galí and Gertler note that they can write their moment condition in many ways, for example the HNKPC could be estimated from either of the following moment conditions:

$$E_t\left[((\theta + \omega(1-\theta(1-\beta)))\pi_t - (1-\omega)(1-\theta)(1-\beta\theta)s_t - \beta\theta\pi_{t+1} - \omega\pi_{t-1})\, z_t\right] = 0 \tag{19}$$

$$E_t\left[\left(\pi_t - \frac{(1-\omega)(1-\theta)(1-\beta\theta)}{\theta + \omega(1-\theta(1-\beta))}s_t - \frac{\beta\theta}{\theta + \omega(1-\theta(1-\beta))}\pi_{t+1} - \frac{\omega}{\theta + \omega(1-\theta(1-\beta))}\pi_{t-1}\right) z_t\right] = 0 \tag{20}$$

Estimation based on these two moment conditions gives surprisingly different results. In particular, (19) leads to an estimate of ω of 0.265 with a standard error of 0.031, but (20) leads to an estimate of 0.486 with a standard error of 0.040. If the model is correctly specified and well-identified, the two equations should, asymptotically, give the same estimates. The fact that the estimates differ suggests that either the model is misspecified or not well identified.

2.4.1 Analyzing Identification

There's an old literature about analyzing identification conditions in rational expectations models. Pesaran (1987) is the classic paper that everyone seems to cite, but I have not read it. Anyway, the idea is to solve the rational expectations model (18) to write it as an autoregression, write down a model for s_t to complete the system, and then analyze identification using familar SVAR or simultaneous equation tools. I will follow Mavroeidis (2005). Another paper that does this is Nason and Smith (2002). Solving (18) and writing an equation for s_t gives a system like:

$$\pi_t = D(L)\pi_{t-1} + A(L)s_t + \epsilon_t \tag{21}$$

$$s_t = \rho(L)s_{t-1} + \phi(L)\pi_{t-1} + v_t \tag{22}$$

$D(L)$ and $A(L)$ are of order the maximum of 1 and the order of $\rho(L)$ and $\phi(L)$ respectively. An order conditions for identification is that the order of $\rho(L)$ plus $\phi(L)$ is at least two, so that you have at least two valid instruments to instrument for s_t and π_{t+1} in (18). This condition can be tested by estimating (22) and testing whether the coefficients are 0. Mavroeidis does this and finds a p-value greater than 30%, so non-identification is not rejected. Mavroeidis then picks a wide range of plausible values for the parameters in the model and calculates the concentration parameter for these parameters. He finds that concentration parameter is often very close to zero. Recall from 382 that in IV, a low concentration parameter indicates weak instrument problems.

2.4.2 Weak Identification in GMM

As with IV, when a GMM model is weakly identified, the usual asymptotic approximations work poorly. Fortunately, there are alternative inference procedures that perform better.

GMM Bias [3] The primary approaches are based on the CUE (continuously updating estimator) version of GMM. To understand why, it is useful to write down the approximate finite sample bias of GMM. If our moment conditions are $g(\beta) = \sum g_i(\beta)/T$ and $\Omega(\beta) = E[g_i(\beta)g_i(\beta)']$ (in the iid case, for time series replace with an appropriate auto-correlation consistent type estimator) CUE minimizes:

$$\hat{\beta} = \arg\min g(\beta)'\Omega(\beta)^{-1}g(\beta)$$

That is, rather than plugging in a preliminary estimate of β to find the weighting matrix, CUE continuously updates the weighting matrix as a function of β. Suppose we used a fixed weighting matrix, A and do GMM. What is the expectation of the objective function? Well, for iid data (if observations are correlated, we will get an even worse bias) we have:

$$\begin{aligned} E\left[g(\beta)'Ag(\beta)\right] =& E\left[\sum_{i,j} g_i(\beta)'Ag_j(\beta)/T^2\right] \\ =& \sum_{i\neq j} E[g(\beta)]AE[g(\beta)]/T^2 + \sum_i E[g_i(\beta)'Ag_i(\beta)]/n^2 \\ =& (1-T^{-1})E[g(\beta)]AE[g(\beta)] + tr(A\Omega(\beta))T^{-1} \end{aligned}$$

The first term is the population objective function, so it is minimized at β_0. The second term, however, is not generally minized at β_0, causing $E[\hat{\beta}_T] \neq \beta$. However, if we use $A = \Omega(\beta)^{-1}$, then the second term vanishes and we have an unbiased estimator. This is sort of what CUE does. It is not exactly since we use $\hat{\Omega}(\beta)$ instead of $\Omega(\beta)$. Nonetheless, it can be shown to be less biased than two-step GMM. See Newey and Smith (2004).

Another view of the bias can be obtained by comparing the first order conditions of CUE and two-step GMM. The first order condition for GMM is

$$0 = G(\beta)\hat{\Omega}(\tilde{\beta})^{-1}g(\beta) \tag{23}$$

[3] This section is based on Whitney's notes from 386.

where $G(\beta) = \frac{\partial g}{\partial \beta} = \sum \frac{\partial g_i}{\partial \beta}$ and $\tilde{\beta}$ is the first step estimate of β. This term will have bias because the ith observation in the sum used for G, $\hat{\Omega}$, and g will be correlated. Compare this to the first order condition for CUE:

$$\begin{aligned} 0 =& G(\beta)\hat{\Omega}(\beta)^{-1}g(\beta) - g(\beta)\hat{\Omega}(\beta)^{-1}\left(\sum\left(\frac{\partial g_i}{\partial \beta}g_i(\beta)' + g_i(\beta)\frac{\partial g_i}{\partial \beta}'\right)/T\right)\hat{\Omega}(\beta)^{-1}g(\beta) \\ =& \left[G(\beta) - \left(\sum \frac{\partial g_i}{\partial \beta}g_i(\beta)'\right)\hat{\Omega}(\beta)^{-1}\right]\hat{\Omega}(\beta)^{-1}g(\beta) \end{aligned}$$

The term in brackets is the projection of $G(\beta)$ onto the space orthogonal to $g(\beta)$. Hence, the term in brackets is uncorrelated with $g(\beta)$. This reduces bias.[4]

Identification Robust Inference The lower bias of CUE suggests that inference based on CUE might be more robust to small sample issues than traditional GMM inference. This is indeed the case. Stock and Wright (2000) showed that under $H_0 : \beta = \beta_0$ the CUE objective function converges to a χ^2_m where m is the number of moment conditions. Moreover, this convergence occurs whether the model is strongly, weakly[5], or non-identified. Some authors call the CUE objective function the S-statistic. Others call it the AR-statistic because in linear models, the AR statistic is the same as the CUE objective function. The S-stat has the same properties as the AR-stat discussed in 382. Importantly, its degrees of freedom grows with the number of moments, so it may have lower power in very over identified models. Also, an S-stat test may reject either because $\beta \neq \beta_0$ or because the model is misspecified. This can lead to empty confidence sets.

The Kleibergen (2005) developed an analog of the Lagrange Multiplier that, like the S-stat, has the same limiting distribution regradless of identification. The LM stat is based on the fact that under $H_0 : \beta = \beta_0$, the derivative of the objective function at β should be approximately zero. Kleibergen applies this principal to the CUE objective function. Let $\hat{D}(\beta) = G(\beta) - \widehat{acov}\left(G(\beta), g(\beta)\right)\hat{\Omega}(\beta)^{-1}$ (as above for iid data, $\widehat{acov}\left(G(\beta), g(\beta)\right) = \sum \frac{\partial g_i}{\partial \beta}g_i(\beta)'$). Kleibergen's statistic is

$$KLM = g(\beta)\Omega(\hat{\beta})^{-1}\hat{D}(\beta)(\hat{D}(\beta)'\Omega(\hat{\beta})^{-1}\hat{D}(\beta))^{-1}\hat{D}(\beta)'\Omega(\hat{\beta})^{-1}g(\beta) \xrightarrow{d} \chi^2_p \tag{24}$$

It is asymptotically χ^2 with p =(number of parameters) degrees of freedom. The degrees of freedom of KLM does not depend on the degree of overidentification. This can give it better power properties than the AR/S stat. However, since it only depends on the first order condition, in addition to being minimized at the minimum of the CUE objective function, it will also be minimized at local minima and maxima and inflection points. This property leads Kleibergen to consider an identification robust version of the Hansen's J-statistic for testing overidentifying restriction. Kleibergen's J is

$$J(\beta) = S(\beta) - KLM(\beta) \xrightarrow{d} \chi^2_{m-p} \tag{25}$$

Moreover, J is asymptotically independent of KLM, so you can test using both of them, yielding a joint test with size $\alpha = \alpha_J + \alpha_K - \alpha_J\alpha_K$.

If you have a great memory, you might also remember Moreira's conditional likelihood ratio test from covering weak instruments in 382. There's also a GMM version of this test discussed in Kleibergen (2005).

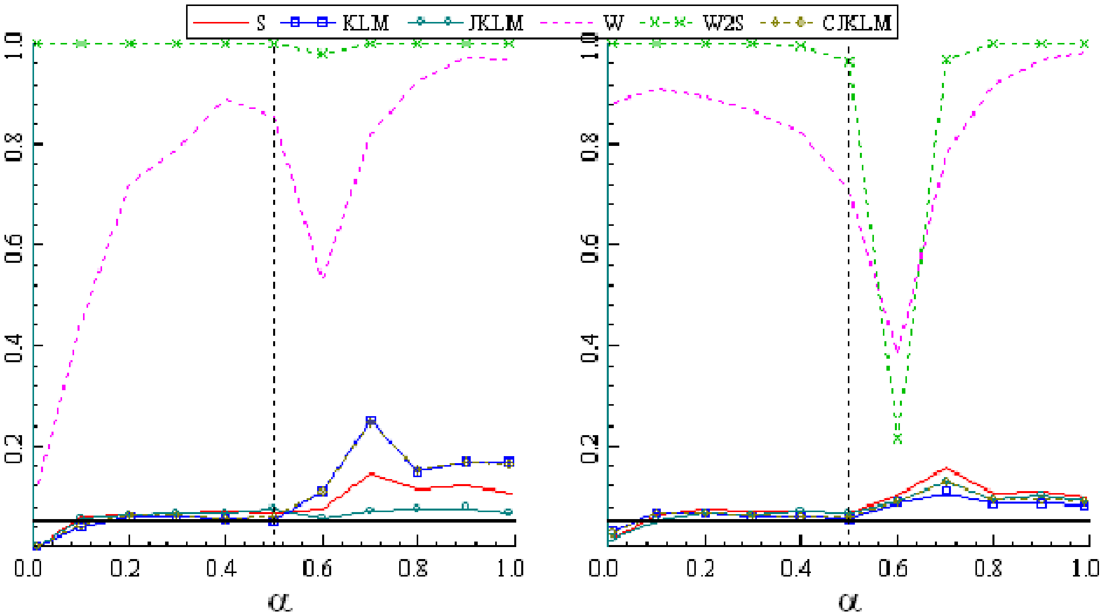
2.4.3 Results of Weak Identification Robust Inference for HNKPC

Kleibergen and Mavroeidis Kleibergen and Mavroeidis (2008) extend Kleibergen's tests described above, which only work for testing a the full set of parameters, to tests for subsets of parameters. As an application, Kleibergen and Mavroeidis (2008) simulate a HNKPC model and consider testing whether the faction of backward looking firms (which they call α, but GG and I call ω) equals one half. Figure 1 shows the frequency of rejection for various true values of α. The Wald test badly overrejects when the true α is 1/2. The KLM and JKLM have the correct size under H_0, but they also have no power against any of the alternatives. It looks like identification is a serious issue.

[4] There is still some bias due to parts of $\hat{\Omega}$ being correlated with g and G.

[5] Defining weak GMM asymptotics involves introducing a bunch of notation, so I'm not going to go through it. The idea is essentially the same as in linear models. See Stock and Wright (2000) for details.

Figure 1: Kleibergen and Mavroeidis Power Curve



Dufour, Khalaf, and Kichian (2006 Use the AR and K statistics to construct confidence sets for Galí and Gertler's model. Figure 2 shows the results. The confidence sets are reasonably informative. The point estimates imply an average price duration of 2.75 quarters, which is much closer to the micro-data evidence (Bils and Klenow's average is 1.8 quarters) than Galí and Gertler's estimater. Also, although not clear from this figure, Dufour, Khalaf, and Kichian find that Galí and Gertler's point estimates lie outside their 95% confidence sets.

Figure 2: Dufour, Khalaf, and Kichian Confidence Sets