Simulation Estimators

Summer School in Structural Estimation

Toni M. Whited

June 2023

Outline

Pseudo-Random Numbers

- Computers cannot produce truly random numbers, and instead produce what are called "pseudo-random" numbers.
- In general, pseudo-random numbers appear random because they can pass some simple tests of randomness such as a test for serial correlation.
- A very simple example of a pseudo-random number generator is

$$X_r = (kX_{r-1} + c)\,\mathrm{mod}\,m$$

  in which the modulo operator $a\,\mathrm{mod}\,b$ produces the remainder of a/b.
- Actual pseudo-random number generators are more complicated, but they use the same basic principle.

Pseudo-Random Numbers

- A series of (pseudo) random $i.i.d.$ uniformly distributed random numbers on $(0, m)$ $u_r, u_{r+1}, \ldots, r_R$ can be produced as $u_r = X_r/m$.
- Most statistical packages have functions that produce random uniform and random normal vectors.
- These random numbers are in reality deterministic. The periodicity of the cycle is determined by the k, c, and m values.
- For well chosen values, the period is typically high (e.g. 10^9).
- X_0 is referred to as the seed or state number, which determines the sequence of random numbers.
- To use the same sequence of random numbers repeatedly, remember to either save the random number draws from the first use or re-use the same seed number.

Outline

What if . . . you wanted to estimate a mean?

- Suppose you have an $i.i.d.$ sample, y_i, of length N.
- You could use classical method-of-moments by picking an estimate, μ, based on the following moment condition:
$$E(y-\mu)=0$$
- The sample counterpart to this moment condition is
$$\frac{1}{N}\sum_{i=1}^{N} y_i - \hat{\mu} = 0$$
- In this example you have a closed-form expression for the moment condition.
- But you might not . . .

What if . . . you wanted to estimate a mean?

- A different, very convoluted, way to do the same thing would be to proceed as follows.
  - Generate a random vector with a mean, μ. Calculate its average. Call the average from this simulation $\hat{\mu}_1$.
  - Do this S times and then calculate.

$$\tilde{\mu} = \frac{1}{S}\sum_{i=1}^{S}\hat{\mu}_i$$

  - Pick the estimate, μ that sets

$$\tilde{\mu} - \frac{1}{N}\sum_{i=1}^{N} y_i$$

as close to zero as possible.

What if . . . you wanted to estimate a mean?

- This is a simulated method of moments estimator.
- In practice, the estimator is almost the same because you simulate a random vector with a mean μ by simulating a random vector with a mean of zero and then adding μ.
- Its variance will not be the same as the variance of a GMM estimator because of simulation error.
- This example is, of course, silly because we don't need to calculate μ via a simulation. We know it. We just write it down.
- However, there exist applications in which this is not the case.

Outline

SMM is a special case of GMM, so . . .

- Let's briefly review GMM.

- We will come back to this material later in much more detail when we talk about weight matrices for SMM

The Setup

- The following uses the notation in Wooldridge.
- Let
  - Let $\boldsymbol{w}_i$ be an $(M \times 1)$ be an $i.i.d.$ vector of random variables for observation i.
  - $\boldsymbol{\theta}$ be an $(P \times 1)$ vector of unknown coefficients.
  - $\boldsymbol{g}(\boldsymbol{w}_i, \boldsymbol{\theta})$ be an $(L \times 1)$ vector of functions $\boldsymbol{g} : \left(\mathcal{R}^M \times \mathcal{R}^P\right) \to \mathcal{R}^L, \quad L \geq P$
- The function $\boldsymbol{g}(\boldsymbol{w}_i, \boldsymbol{\theta})$ can be nonlinear.
- Let $\boldsymbol{\theta}_0$ be the true value of $\boldsymbol{\theta}$.
- Let $\hat{\boldsymbol{\theta}}$ represent an estimate of $\boldsymbol{\theta}$.
- The “hat” and “naught” notation applies to anything we might want to estimate.

Moment Restrictions

- GMM is based on what are generally called moment restrictions and sometimes called orthogonality conditions (The latter terminology comes from the rational expectations literature.)

$$E\left(\boldsymbol{g}\left(\boldsymbol{w}_i, \boldsymbol{\theta}_0\right)\right) = 0$$

- This condition is expressed in terms of the population. The corresponding sample moment restriction is

$$\frac{1}{N}\sum_{i=1}^{N} \boldsymbol{g}\left(\boldsymbol{w}_i, \boldsymbol{\theta}\right) = 0$$

- What we want to do is choose $\hat{\boldsymbol{\theta}}$ to get $N^{-1}\sum_{i=1}^{N} \boldsymbol{g}\left(\boldsymbol{w}_i, \boldsymbol{\theta}\right)$ as close to zero as possible.

Criterion Function

- The estimator, $\hat{\boldsymbol{\theta}}$ minimizes a quadratic form:

$$\boldsymbol{Q}_N(\boldsymbol{\theta}) = \underset{(1\times L)}{\left[N^{-1}\sum_{i=1}^{N}\boldsymbol{g}(\boldsymbol{w}_i,\boldsymbol{\theta})\right]'} \underset{(L\times L)}{\widehat{\boldsymbol{\Xi}}} \underset{(L\times 1)}{\left[N^{-1}\sum_{i=1}^{N}\boldsymbol{g}(\boldsymbol{w}_i,\boldsymbol{\theta})\right]}$$

  where $\widehat{\boldsymbol{\Xi}}$ is a positive definite matrix that converges in probability to $\boldsymbol{\Xi}_0$

- In this case, $\boldsymbol{Q}_N$ converges in probability to

$$\{E[\boldsymbol{g}(\boldsymbol{w}_i,\boldsymbol{\theta})]\}'\boldsymbol{\Xi}\{E[\boldsymbol{g}(\boldsymbol{w}_i,\boldsymbol{\theta})]\}$$

Outline

Setup

- Let w_i be an $i.i.d.$ data vector, $i = 1, \ldots, n$.
- Let $y_{is}(\theta)$ be an $i.i.d.$ simulated vector from simulation s, $i = 1, \ldots, n$, and $s = 1, \ldots, S$.
- The simulated data vector, $y_{is}(\theta)$, depends on a vector of structural parameters, θ.
- The goal is to estimate θ by matching a set of *simulated moments*, denoted as $g(y_{is}(\theta))$, with the corresponding set of actual *data moments*, denoted as $\mathbb{E}(g(w_i))$.
- The simulated moments, $g(y_{is}(\theta))$ are functions of the parameter vector θ because the moments will differ depending on the choice of θ.

Moment Matching

- The first step is to estimate $\mathbb{E}\left(g\left(w_i\right)\right)$ using the actual data.
- The second step is to construct S simulated data sets based on a given parameter vector.
- For each of these data sets, compute a simulated moment, $g\left(y_{is}\left(\theta\right)\right)$.
- Note that you have to make the *exact* same calculations on the simulated data as you do on the real data.
- $Sn > n$.
- Michaelides and Ng (2000) find that good finite sample performance requires a simulated sample that is approximately ten times as large as the actual data sample.

Now let's figure out how to match the moments

- Define
$$g_n(\theta) = n^{-1} \sum_{i=1}^{n} \left[g(w_i) - S^{-1} \sum_{s=1}^{S} g(y_{is}(\theta)) \right].$$

- The simulated moments estimator of θ is then defined as the solution to the minimization of
$$\hat{\theta} = \arg\min_{\theta} Q(\theta, n) \equiv g_n(\theta)' \hat{W}_n g_n(\theta),$$

- $\hat{W}_n$ is a positive definite matrix that converges in probability to a deterministic positive definite matrix W.

Weight Matrix

- In most applications, one can use the **optimal** weight matrix, which we will call Λ, and which can be calculated **easily** as the inverse of the variance covariance matrix of $\sum_{i=1}^{n} g(\boldsymbol{w}_i)$.
- **This weight matrix can be calculated without any knowledge of the model!!!**
- Either use GMM or stack influence functions.
  - Computationally identical.
  - We will do influence functions tomorrow!

Weight Matrix

- In many applications one can calculate the weight matrix as the inverse of the variance covariance matrix of $n^{-1}\sum_{i=1}^{n} g(w_i)$. This weight matrix has an exact analogy with GMM.
- In some cases this type of weight matrix is not feasible; for example, in the multinomial probit model.
- In such cases you use a two stage procedure.
  - In the first stage, minimize $Q(\theta, n)$ using the identity as the weight matrix.
  - Use the resulting estimate, $\hat{\theta}$ to construct the weight matrix as the inverse of the variance of g_n.

Inference

- The simulated moments estimator is asymptotically normal for fixed S!! (This is not the case for SMLE.)
- The asymptotic distribution of θ is given by

$$\sqrt{n}\left(\hat{\theta}-\theta\right) \xrightarrow{d} \mathcal{N}\left(0, \operatorname{avar}(\hat{\theta})\right)$$

  in which

$$\operatorname{avar}(\hat{\theta}) \equiv \left(1+\frac{1}{S}\right)\left[\frac{\partial g_n\left(\theta\right)}{\partial \theta} W \frac{\partial g_n\left(\theta\right)}{\partial \theta'}\right]^{-1}.$$

  and W is the efficient weight matrix
- Note larger S (more simulated samples) $\Rightarrow$ smaller standard errors

Inference (numerical derivative)

In previous formula, how to compute Jacobian $\partial g_n(\theta)/\partial\theta$?

1. Get the SMM estimate, $\hat{\theta}$
2. Remember the dimensions:
   - Parameters θ: $k \times 1$
   - Moments g: $m \times 1$
   - Jacobian $\partial g_n(\theta)/\partial\theta$: $k \times m$
3. Choose step size $h_i > 0$ for each parameter $i = 1, ..., k$
   - Each parameter gets its own step size
   - Must be chosen with care! Smaller is better ... except when it's not
   - Good initial guess: 1% of parameter's estimate
4. Compute two-sided difference. Element $\{i, j\}$ of $\partial g_n(\theta)/\partial\theta$ is

$$\frac{\partial g_{n,j}\,(\theta)}{\partial\theta_i} \approx \frac{g_{n,j}(\hat{\theta} + h_i) - g_{n,j}(\hat{\theta} - h_i)}{2h_i},$$

where "$+h_i$" means perturb parameter i upward by h_i

Test of Overidentifying Restrictions

- As in the case of plain vanilla GMM, one can perform a test of the overidentifying restrictions of the model

$$\frac{nS}{1+S}Q(\theta, n)$$

- This statistic converges in distribution to a χ^2 with degrees of freedom equal to the dimension of g_n minus the dimension of θ.

So How Do you Actually DO SMM? Data Step

- The data steps:
  - Choose moments to match. Can include means, variances, covariances, regression slopes, etc. Need at least as many moments as parameters.
  - Extra moments provide a test of overidentifying restrictions.
  - Compute moments in the actual data and stack them in a vector
  - Estimate the covariance matrix of this vector. Invert it. This is your efficient SMM/GMM weighting matrix.
  - Calculate many more moments than you think you will need. You **will** change your moment vector at some point.

So How Do You Actually DO SMM?

- Choose an optimizer. These usually have two inputs, a set of parameters over which to optimize and a function to optimize.
- Write a function to be optimized. It will input a parameter vector and output a GMM objective function. This will involve reading in data moments and a weight matrix and then calculating simulated moments and then forming a quadratic form.
- The function subroutine will have to call a model solving routine, a model simulating routine, and a moment calculating routine. The goal in this chain is to eat up parameters and spit back moments.
- When you are done, calculate the Jacobian matrix and the standard errors.

Pseudo Code

```
function SMM_Objective_Function(in double parameters[numberParameters], out double objectiveFunctionValue)
        read weightMatrix[numberMoments, numberMoments]
        read dataMoments[numberMoments]
        call function solveTheModel( in double parameters[numberParameters],
                out double valueFunction[stateSpaceSize],
                out int policyFunction[stateSpaceSize])
        call function simulateFirms( in double valueFunction[stateSpaceSize],
                in int policyFunction[stateSpaceSize],
                out double simulatedFirms[numberOfFirms, numberOfVariables])
        call function calculateMoments(in double simulatedFirms[numberOfFirms, numberOfVariables],
                out double SimulatedMoments[numberMoments])
        momentError = dataMoments − SimulatedMoments
        objectiveFunctionValue = momentError′ * weightMatrix * momentError
```

Outline

You need to worry about identification

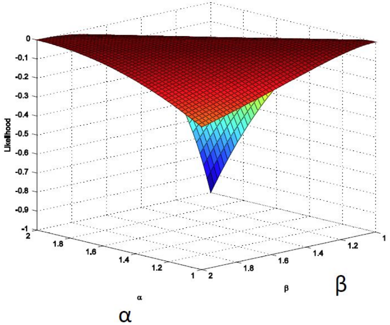
- Identification means different things in different contexts. Sigh.
- What does “identification” usually mean in reduced-form work?
  - Does x affect y?
  - Does y affect x?
  - Does some omitted variable z affect both x and y?
- Exogenous variation is very useful for answering this kind of question
- “Identification” in these papers usually means that the researcher has an experimental design that establishes causality and is identifying a direction.

We will use the formal statistical definition of identification

- Econometrician defines an objective function over parameters and data
- Goal: Select parameter values that minimize this objective function
  - Example: Find slope coefficient that minimize sum of squared errors
- A parameter is (point) *identified* if there is a unique minimum for the objective function at parameter's true value in the population
- In what follows, I'll use the formal statistical definition of identification

Example of an unidentified model

- Suppose we want to estimate parameters α and β by MLE
- Parameters α and β appear in the likelihood function only in the form α/β
- $\Rightarrow \alpha/\beta$ is identified, but α and β are not separately identified
- Likelihood function is flat at its max:



Identification and SMM

- The success of the SMM procedure relies on picking moments g that can identify the structural parameters θ
- The conditions for global identification of a simulated moments estimator are similar to those for GMM:
  - The expected value of the difference between the simulated and actual moments equals zero iff the structural parameters are at their true value
  - A sufficient condition for identification is a one-to-one mapping between the structural parameters and a subset of the data moments of the same dimension

Identification and SMM

- Let $g_\theta\left(y_{is}\left(\theta\right)\right)$ be a subvector of $g\left(y_{is}\left(\theta\right)\right)$ with the same dimension as θ
- Local identification implies that the Jacobian determinant,

$$\det\left(\partial g_\theta\left(y_{is}\left(\theta\right)\right)/\partial\theta\right),$$

  is non-zero. i.e., Jacobian has full rank.
- This condition can be interpreted loosely as saying that the moments, g_θ, are informative about the structural parameters, θ
- That is, the sensitivity of moments to parameters is high

How to choose moments in SMM/GMM

- Best-case scenario: Each moment depends on just one model parameter
  - "Moment #1 identifies parameter #1, moment #2 identifies parameter #2..."
  - This is how many macro papers are (misleadingly) written
- More realistic: Every moment depends on every parameter
- All parameters can affect all moments, but the mapping has to be one-to-one and onto
- Do comparative statics to understand how each moment moves with each parameter. Make sure you understand the economics behind each comparative static.
- Need enough moments, and moments that move in different directions for different parameters, to obtain identification
- Try targeting empirical policy functions (more details later)

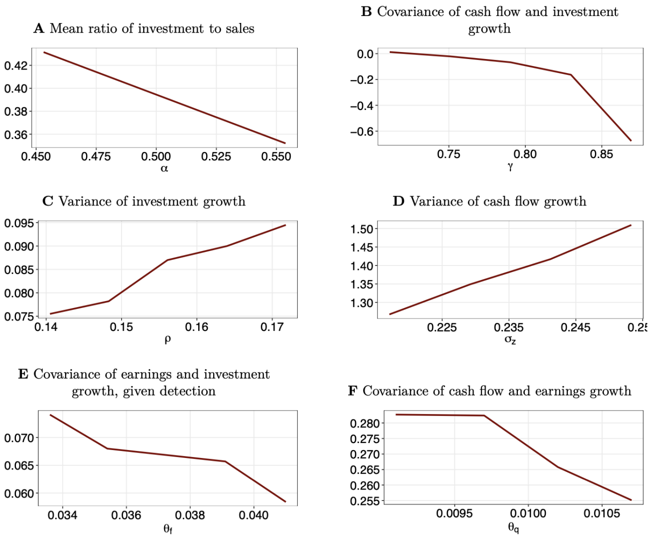
Identification

- Don't start estimating until you're sure the model is identified, or you are in for a world of grief.
- It's usually impossible to prove formally that your model is identified
- How do you ensure that the model is identified?

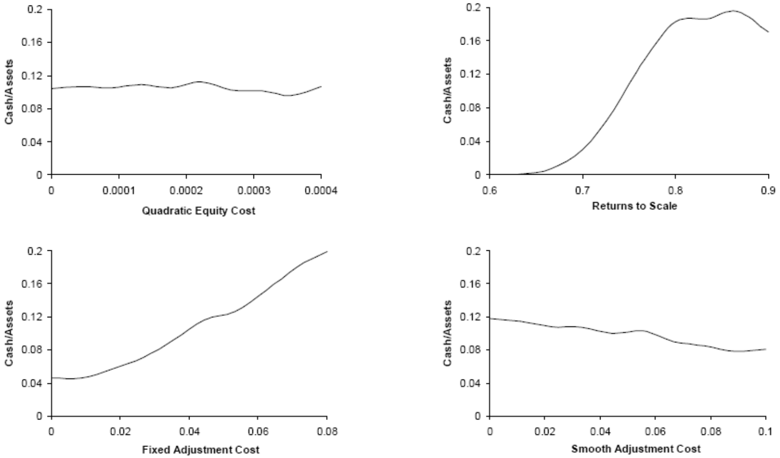
Use Economics (more on this later!)

- **PLAY WITH YOUR MODEL UNTIL YOU UNDERSTAND HOW IT WORKS!!!!!!!!!**
- Do comparative statics: plot the simulated moments as functions of the parameters.
- You want to find steep, monotonic relationships.
- You want moments that move in different directions for different parameters.

Example One: Terry, Whited and Zakolyukina (2023)



Example Two: Riddick and Whited (2009)



Outline

Dynamic models require the estimation of dynamics

- A large fraction of dynamic models have driving processes that follow autoregressive processes.
- One statistic that needs to be matched in many structural estimations is an $AR(1)$ coefficient.

Consistent estimation of a first-order autoregressive coefficient with fixed effects

- Suppose you have a variable

$$y_{it}, \quad i = 1, \ldots, N, \quad t = 1, \ldots, T.$$

  that follows a process

$$y_{it} = \alpha_i + \rho y_{it-1} + u_{it},$$

  in which u_{it} is possibly correlated with α_i.

- OLS will not work.
- You cannot do firm-level deviations from means with a lagged dependent variable.
- Dynamic panel models suffer from weak instrument problems.

Han and Phillips (2010)

- Han and Phillips (2010) use double differencing to remove the fixed effect.
- Some ugly but easy algebra shows that a consistent estimate of ρ is given by

$$\hat{\rho} = \frac{\sum_{i=1}^{N} \sum_{t=2}^{T} \Delta y_{it-1} \left(2\Delta y_{it} + \Delta y_{it-1}\right)}{\sum_{i=1}^{N} \sum_{t=2}^{T} \left(\Delta y_{it-1}\right)^{2}}$$

  where Δ is the first difference operator.
- This estimator is clearly obtained from regressing $2\Delta y_{it} + \Delta y_{it-1}$ on Δy_{it-1}.

Sample Julia code for a balanced panel

Consider a panel with dimensions T and N:

```
k = size(biginflnc,2);
vmx = zeros(k,k);
for ic in 1:capN;
    global phii = biginflnc[(ic-1)*capT+1:ic*capT,:];
    phii = sum(phii,dims=1);
    global vmx = vmx + phii'*phii;
end

vmx = vmx./((capN*capT)^2);
```

Han, C.„ Phillips, P.C.B., 2010. GMM estimation for dynamic panels with fixed effects and strong instruments at unity. Econometric Theory 26, 119–151.

Michaelides, A.„ Ng, S., 2000. Estimating the rational expectations model of speculative storage: A Monte Carlo comparison of three simulation estimators. Journal of Econometrics 96, 231–266.

Riddick, L.A.„ Whited, T.M., 2009. The corporate propensity to save. Journal of Finance 64, 1729–1766.

Terry, S.J., Whited, T.M.„ Zakolyukina, A., 2023. Information versus investment. Review of Financial Studies 36, 1148–1191.