Summer School in Structural Estimation
Simulation Estimators

University of Michigan
Ross School of Business

Toni M. Whited

August 2021

Outline

- Today
  - An aside on random numbers
  - An outline of simulated moments and simulated minimum distance estimation
- Tomorrow
  - Making weight matrices with influence functions
  - Advice from the trenches and applied papers

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

$$X_r = (kX_{r-1} + c) \bmod m$$

  in which the modulo operator $a \bmod b$ 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.

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:
$$E\left(y-\mu\right)=0$$
- The sample counterpart to this moment condition is
$$\frac{1}{N}\sum_{i=1}^{N}y_i-\mu=0$$
- 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.

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.
  - $\boldsymbol{g}\left(\boldsymbol{w}_i, \boldsymbol{\theta}\right)$ be an $(L \times 1)$ vector of functions $\boldsymbol{g}:\left(\mathcal{R}^M \times \mathcal{R}^P\right) \rightarrow \mathcal{R}^L, \quad L \geq P$
- 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

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

Setup

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

Moment Matching

- The first step is to estimate $h(\boldsymbol{w}_i)$ 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, estimate a simulated moment, $h(y_{is}(\boldsymbol{\theta}))$.
- Note that you have to make the *exact* same calculations on the simulated data as you do on the real data.
- SN need not equal 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

- In Wooldridge's notation, define

$$\boldsymbol{g}(\boldsymbol{w}_i, \boldsymbol{\theta}) \equiv h\left(\boldsymbol{w}_i\right) - S^{-1}\sum_{s=1}^{S} h\left(y_{is}\left(\boldsymbol{\theta}\right)\right)$$

- The simulated moments estimator of $\boldsymbol{\theta}$ is then defined as the solution to the minimization of

$$\hat{\boldsymbol{\theta}} = \arg\min_{\theta} \boldsymbol{Q}_n(\boldsymbol{\theta}) \equiv \boldsymbol{g}(\boldsymbol{w}_i, \boldsymbol{\theta})' \hat{\boldsymbol{\Xi}}_n \boldsymbol{g}(\boldsymbol{w}_i, \boldsymbol{\theta}),$$

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

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 $h(\boldsymbol{w}_i)$.
- **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 some cases this type of weight matrix is not feasible (multinomial probit).
- In such cases you use a two stage procedure.
  - In the first stage, minimize $Q_n(\theta)$ 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 $\sqrt{N}h\left(y_{is}\left(\theta\right)\right)$. This computation entails that you re-solve your model and re-simulate your data many times.

Inference

- The simulated moments estimator is asymptotically normal for fixed S!! (This is not the case for SMLE.)
- Let $\boldsymbol{G}_0 = E\left[\dfrac{\partial \boldsymbol{g}\left(\boldsymbol{w}_i, \boldsymbol{\theta}\right)}{\partial \boldsymbol{\theta}}\right]$.
- The asymptotic distribution of $\boldsymbol{\theta}$ is given by

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

in which

$$\operatorname{avar}(\hat{\boldsymbol{\theta}}) \equiv \left(1 + \frac{1}{S}\right)\left[G_0 \Lambda G_0'\right]^{-1}.$$

Inference

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

$$\frac{nS}{1+S}Q_n(\boldsymbol{\theta})$$

- This statistic converges in distribution to a χ^2 with degrees of freedom equal to the dimension of g_n minus the dimension of $\boldsymbol{\theta}$.

Identification

- The success of this procedure relies on picking moments h that can identify the structural parameters $\boldsymbol{\theta}$.
- 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 moments and the data moments equals zero iff the structural parameters equal their true values.
  - 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

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

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

  is non-zero.
- This condition can be interpreted loosely as saying that the moments, h, are informative about the structural parameters, $\boldsymbol{\theta}$.
- That is, the sensitivity of h to $\boldsymbol{\theta}$ is high.
- Picking good moments is analogous to picking strong instruments in a standard IV estimation.

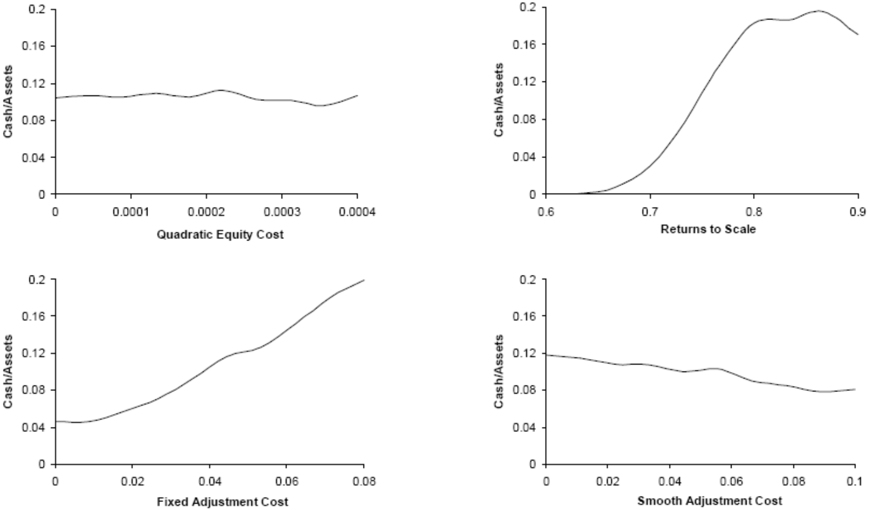
Identification

- How do you ensure that the model is identified?
- Do an informal numerical check.
- Check the standard errors:
  - The precision of the estimates, measured through the asymptotic variance above, is related to the sensitivity of the auxiliary parameters to movements in the structural parameters through $\partial h\left(y_{is}\left(\boldsymbol{\theta}\right)\right)/\partial\boldsymbol{\theta}$
  - If the sensitivity is low, the derivative will be near zero, which will produce a high variance for the structural estimates.

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.

Examples[1]

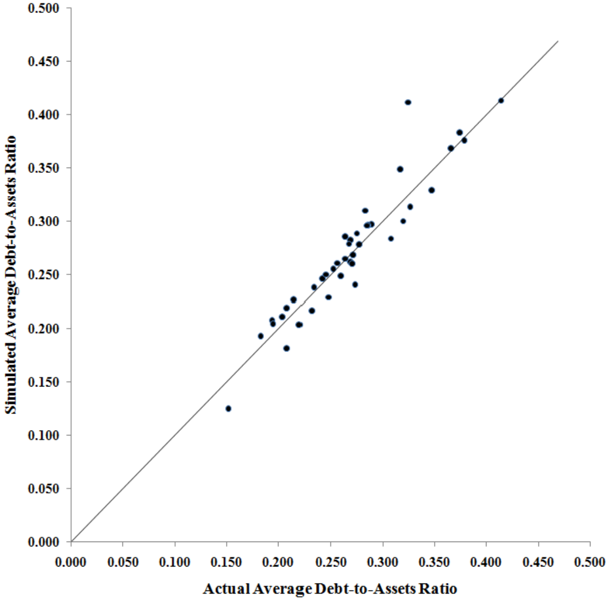


[1] Riddick and Whited (2009)

Firm Heterogeneity

- It is really hard to address the issue of firm heterogeneity using SMM.
- This is perhaps the biggest drawback of this technique.
- The models we simulate are usually of a single firm or at best an industry equilibrium with very limited heterogeneity.
- So you have to suck as much heterogeneity out of your data as you can before you can have any hope of fitting the model to the data.
  - Firm and time fixed effects
- You can also do sample splits. This used to be computationally infeasible, but . . .

Sample Splits[2]



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.

Logistics

- How on earth do you minimize something that does not have a closed form?
- For any given value of $\boldsymbol{\theta}$, say $\boldsymbol{\theta}_0$.
  - Solve your model.
  - Simulate $y_{is}(\boldsymbol{\theta})$ and compute $h(y_{is}(\boldsymbol{\theta}))$.
  - Evaluate the objective function, $\boldsymbol{Q}_n(\boldsymbol{\theta})$.
- Choose a new value for the parameters, say $\boldsymbol{\theta}_1$, for which $\boldsymbol{Q}(\boldsymbol{\theta}_0, n) > \boldsymbol{Q}(\boldsymbol{\theta}_1, n)$.

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 gradient matrix and the standard errors.

Pseudo Code

```
function SMM (in double parameters[numberParameters],
                    out double objectiveFunctionValue)
          call function solveTheModel( in double parameters[numberParameters],
                    out double valueFunction[stateSpaceSize],
                    out int policyFunction[stateSpaceSize])
          read weightMatrix[numberMoments, numberMoments]
          read dataMoments[numberMoments]
          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 SiimulatedMoments[numberMoments])
          momentError = dataMoments − SimulatedMoments
          objectiveFunctionValue = momentError * weightMatrix * momentError
```

Economic Models and Auxiliary Models

- If a model does not provide a closed form likelihood, or the simulated likelihood needs to be calculated for more than two variables (and two is a stretch), then you can use an auxiliary model to estimate the model parameters.
- A likelihood is a description of the true data generating process.
- An auxiliary model is an approximation of the true data generating process.
  - What if you have a DSGE model of the macroeconomy. The likelihood is impossible to solve for, but a VAR might describe the data approximately.
  - What if you have a model of investment spikes or infrequent price adjustments. The auxiliary model could be a duration model.

Auxiliary Models

- In practice, the auxiliary model is itself characterized by a set of parameters.
- These parameters can be estimated using either the observed data or the simulated data.
- Indirect inference chooses the parameters of the underlying economic model so that these two sets of estimates of the parameters of the auxiliary model are as close as possible.

Auxiliary Models

- You should be able to match exactly if you have as many parameters in the auxiliary model as you do in the economic model.
- But the number of auxiliary parameters can be greater than the number of economic parameters.
- To the extent that the parameters of the auxiliary model are functions of moments of the data, indirect inference can be thought of as a superset of SMM.
- It falls under the category of a simulated minimum distance (SMD) estimator.

Estimation: Minimum Distance Style

- Notation:
    - w_N is a data matrix of length N
    - w_N^s is a simulated data matrix of length N from simulation s, $s = 1, \ldots, S$.
    - b a vector of **auxiliary** model parameters estimated with **real** data.
    - b^s a vector of **auxiliary** model parameters estimated with **simulated** data.
    - θ is the vector of parameters from the **economic** model.

- Without loss of generality, the parameters of the auxiliary model can be represented as the solution to the maximization of a criterion function

$$b_N = \arg\max_{b} J_N\left(w_N, \theta\right),$$

- Examples?

Estimation: Minimum Distance Style

- Construct S simulated data sets based on a given parameter vector, θ.
- For each of these data sets, estimate b^s by maximizing an analogous criterion function

$$b_N^s(\theta) = \arg\max_b J_N\left(w_N^s, b^s(b)\right),$$

- Note that $b_N^s(\theta)$ is an explicit function of the structural parameters, θ.
- The inverse of this function is what Gouriéroux and Monfort (1996) call a “binding” function.

Estimation: Minimum Distance Style

- The indirect estimator of θ is then defined as the solution to the minimization of

$$\hat{\theta} = \arg\min_{\theta} \left[b_N - \frac{1}{S} \sum_{h=1}^{S} b_N^s(\theta) \right]' \hat{W}_N \left[b_N - \frac{1}{S} \sum_{h=1}^{S} b_N^s(\theta) \right]$$
$$\equiv \arg\min_{\theta} \hat{G}_N' \hat{W}_N \hat{G}_N$$

- $\hat{W}_N$ is a positive definite matrix that converges in probability to a deterministic positive definite matrix W.
- As in GMM the optimal weight matrix is the inverse of the covariance matrix of b.
- The main difference between SMM and this flavor of indirect inference is that the former uses **moments** and the latter uses **functions of moments**.

Inference: Minimum Distance Style

- The indirect estimator is asymptotically normal for fixed S. Define $J \equiv \text{plim}_{N\to\infty}\left(J_N\right)$. Then

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

where

$$\text{avar}(\hat{\theta}) \equiv \left(1+\frac{1}{S}\right)\left[\frac{\partial J}{\partial\theta\partial b'}\left(\frac{\partial J}{\partial b}\frac{\partial J'}{\partial b}\right)^{-1}\frac{\partial J}{\partial b\partial\theta'}\right]^{-1}.$$

- The technique provides a test of the overidentifying restrictions of the model, with

$$\frac{NS}{1+S}\hat{G}_N'\hat{W}_N\hat{G}_N$$

converging in distribution to a χ^2, with degrees of freedom equal to the dimension of g_n minus the dimension of θ.

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}.

DeAngelo, H., DeAngelo, L.„ Whited, T.M., 2011. Capital structure dynamics and transitory debt. Journal of Financial Economics 99, 235–261.

Gouriéroux, C.„ Monfort, A., 1996. Simulation-Based Econometric Methods. Oxford University Press, Oxford, United Kingdom.

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.