Simulation Estimators

AES Summer School in Structural Estimation

Toni M. Whited

July 2020

Why use complicated simulation estimators?

- "Better data and computing facilities, have made sensible things simple.[1]"
- Simulation estimators make transparent the relationship between economic models and the equations used to estimate them.
- It seems odd to call computationally intensive econometric techniques "simple," especially given the all-too-frequent criticism that they are a "black box."
- However, there exists a tension between realism and the sorts of models that can produce closed-form estimating equations.
- Better models that can explain more phenomenon may not lend themselves to closed-form solutions.

[1] Ariel Pakes, Keynote address delivered at the inaugural International Industrial Organization Conference in Boston, April 2003

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

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

  in which the modulo operator $a \text{mod} b$ 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.

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.

Setup

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

Moment Matching

- The first step is to estimate $h(x_i)$ 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, $h(y_{is}(b))$.
- 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

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

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

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

Weight Matrix

- In many applications one can calculate the weight matrix as the inverse of the variance covariance matrix of $h(x_i)$. 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 $Q(b, n)$ using the identity as the weight matrix.
  - Use the resulting estimate, $\hat{b}$ to construct the weight matrix as the inverse of the variance of $\sqrt{N} h(y_{is}(b))$. This computation entails that you resimulate your model/data S times.

Inference

- The simulated moments estimator is asymptotically normal for fixed S!! (This is not the case for SMLE.)

- The asymptotic distribution of b is given by

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

  in which

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

  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 $\partial g_n(b)/\partial b$?

1. Get the SMM estimate, $\hat{b}$
2. Remember the dimensions:
   - Parameters b: $k \times 1$
   - Moments h: $m \times 1$
   - Jacobian $\partial g_n(b)/\partial b$: $k \times m$
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 $\partial g_n(b)/\partial b$ is

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

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

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

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

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.

Logistics

- How on earth do you minimize something that does not have a closed form?
- For any given value of b, say b_0.
  - Solve your model.
  - Simulate $y_{is}(b)$ and compute $h(y_{is}(b))$.
  - Evaluate the objective function, $Q(b, n)$.
- Choose a new value for the parameters, say v_1, for which $Q(b_0, n) > Q(b_1, n)$.

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
```

You need to worry about identification

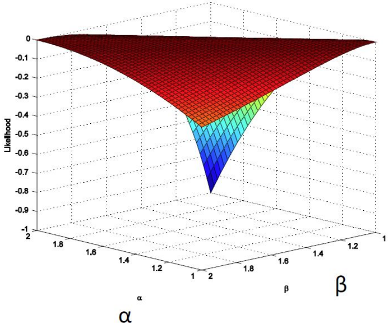
- Identification is often confused with establishing causation
- What does “identification” usually mean in reduced-form work?
  - Does x affect y?
  - Does y affect x?
  - Does some omitted variable z affect both x and y?
- Exogenous variation is very useful for answering this kind of question
- “Identification” in these papers usually means that the researcher has an experimental design that establishes causality

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 by itself is not the ultimate goal

- A parameter can be identified (in the formal statistical sense) without being economically interesting
- Two examples:
  - Regression of endogenous Y on endogenous X
  - Regression of endogenous Y on exogenous X (without a clear and interesting economic mechanism)
- Your ultimate goal: Identify **economically interesting** parameters
  - The parameters may be elasticities defining causal effects
  - But they need not be!
  - **Not all economically interesting parameters are causal elasticities**

"Identification is not causality, and vice versa"

- (Kahn and Whited 2018), Review of Corporate Finance Studies
- Exogenous variation is:
  - Always necessary to identify a causal relation
  - Never sufficient for identifying an economically interesting parameter
    - You also need an economic model (either mathematical or verbal)
  - Only sometimes necessary to identify an economically interesting parameter
- Interesting parameters can sometimes be identified without exogenous variation
  - This is often what's going on in structural corporate finance papers
- Sometimes exogenous variation can be helpful in structural work
  - If part of the economic model can be represented as a regression

Identification and SMM

- The success of the SMM procedure reliefs on picking moments h that can identify the structural parameters b
- 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 $h_b\left(y_{is}\left(b\right)\right)$ be a subvector of $h\left(y_{is}\left(b\right)\right)$ with the same dimension as b
- Local identification implies that the Jacobian determinant,

$$\det\left(\partial h_b\left(y_{is}\left(b\right)\right)/\partial b\right),$$

  is non-zero. I.e., Jacobian has full rank.
- This condition can be interpreted loosely as saying that the moments, h, are informative about the structural parameters, b
- 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 1 model parameter
  - "Moment #1 identifies parameter #1, moment #2 identifies parameter #2. . . "
  - But in this case you wouldn't even need to do SMM/GMM
- 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)

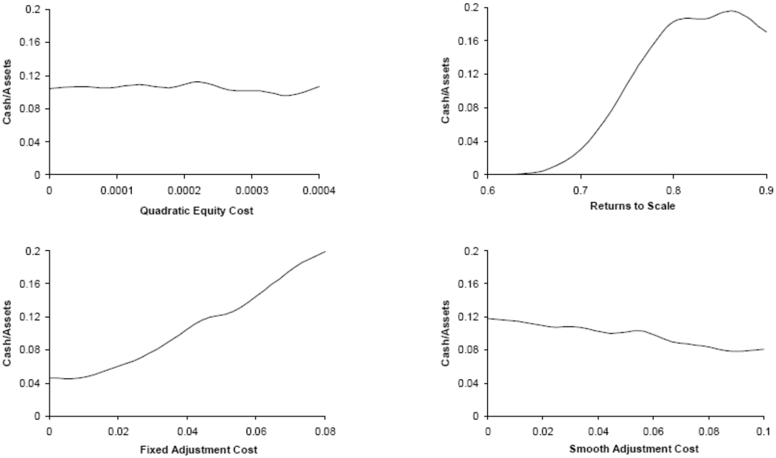
Identification

- 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.

Examples[2]



[2] Riddick and Whited (2009, JF)

Identification checks

- There are several ways to check whether your model is well identified
- Check #1: Check the standard errors
  - Huge standard errors are usually a symptom of poor identification
  - The parameters' standard errors depend on $\partial h_b \left(y_{is} \left(b \right) \right) / \partial b$, the sensitivity of moments to parameters
  - If the sensitivity is low, the derivative will be near zero, which will produce huge standard errors for the structural parameters
- Check #2: Does the Jacobian matrix $\partial h_b \left(y_{is} \left(b \right) \right) / \partial b$ have full rank?
  - If not, your model is not locally identified
  - This is almost the same as a standard error check, but not quite.

Identification checks

- Check # 3: Can estimation recover the true parameter values?
  - Simulate a "fake" dataset off the model
  - Estimate the model, treating the fake data as if it were real data
  - Does the estimator recover the true, known parameter values?
- Check #4: Start searching from different initial parameter guesses
  - Should reach same final estimate regardless of initial guess
  - If not: Coding errors? Stuck in local min? Model not well identified?
- Check #5: Babysit your code
  - Some parameters converge faster than other. Keep track of this.
  - If one or two parameters are changing drastically during the estimation, you have a problem.

Local identification diagnostic

- Andrews, Gentzkow and Shapiro (2017) provide a local diagnostic that measures the sensitivity of $\hat{\theta}$ to the estimated moments, $\hat{g}$.

- It is (and it should look familiar):

$$\boldsymbol{S} = -(\boldsymbol{G}'\boldsymbol{\Xi}\boldsymbol{G})^{-1}\boldsymbol{G}'\boldsymbol{\Xi}.$$

- This measure is not scale invariant, so they propose the following normalization:

$$\boldsymbol{S}_{l,p}\sqrt{\frac{\boldsymbol{\Lambda}_{l,l}}{\boldsymbol{V}_{p,p}}}$$

- I have found this diagnostic to be **very** sensitive to the step size in the numerical gradient routine, but it is certainly worth calculating.

- This diagnostic captures both
  - the steepness of the gradient
  - the precision of the estimation of the moments that identify the parameters

Do you need exogenous variation to do SMM?

- Causal elasticities are not always helpful in structural estimation! (Kahn and Whited 2018)
- Compare SMM estimators of a simple investment model
  - Approach #1: Target moments that measure a causal elasticity
  - Approach #2: Target moments that measure an endogenous elasticity
- Result: The performance of the two approaches is nearly identical!
- In both cases,
  - Multiple moments and multiple parameters interact
  - Identification leans on the structure of the model
- Identification **always** leans on assumptions
  - Even in reduced-form papers with exogenous variation
  - Those papers often lean on a “verbal” economic model

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

$$\theta_N = \arg\max_{\theta} J_N \left(x_N, \theta \right),$$

- Examples?

Estimation: Minimum Distance Style

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

$$\theta_N^s(b) = \arg\max_{\theta} J_N\left(x_N^s, \theta^s(b)\right),$$

- Note that the $\theta_N^s(b)$, as explicit functions of the structural parameters, b.
- The inverse of this function is what Gourieroux and Monfort call a “binding” function.

Estimation: Minimum Distance Style

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

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

- $\hat{W}_N$ 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 θ.
- 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 $J \equiv \text{plim}_{N\to\infty}(J_N)$. Then

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

where

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

- 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 θ minus the dimension of b.

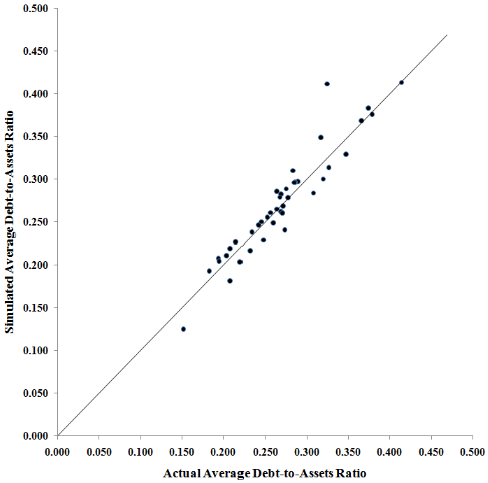
Miscellaneous Pieces of Advice from the Trenches

- Heterogeneity
- The black box problem
- Choosing the weight matrix
- Computation

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[3]



[3] DeAngelo, DeAngelo and Whited (2011)

Do Not Construct a Black Box

- More parameters $\neq$ a better model!!!!!!
- Different features of the data should change when underlying parameters change.
- This is less likely to happen in a model with many extraneous parameters.
- If the author cannot clearly explain which features of the data identify each parameter, the paper / job market candidate is a “reject”
- Structural estimation should not be a black box.

How Not Construct a Black Box

- Start simple!!!
- Make sure the simple model is **right**.
- Figure out where the simple model succeeds and fails.
- Add a feature that will help you answer the question you want to answer.
- Make sure the slightly more complicated model is right.
- Figure out where the more complicated model succeeds and fails.
- Converge.

The question comes first — Not the model

- Before going structural, convince yourself that a structural approach is absolutely necessary.
- Are there data limitations?
- Are you interested in something besides a causal elasticity?

Use the right weighting matrix

- Many structural estimation papers use odd weighting matrices.
- Identity weight matrix
  - This choice mechanically puts more weight on moments that are larger in absolute value.
  - Not an economically sensible choice.
  - Bazdresch, Kahn and Whited (2018) note poor finite sample performance.

Use the right weighting matrix

- Inverse of the squared moments on the diagonal
  - This choice minimizes the percentage differences in the moments.
  - This choice mechanically puts more weight on moments that are smaller in absolute value.
  - Not an economically sensible choice.

Use the right weighting matrix

- Inverse moment variance along the diagonal
  - This choice puts the least weight on moments that are estimated with the least precision.
  - A statistically more or less sensible choice.
  - The old version of Bazdresch et al. (2018) (before the referees got to it) note poor finite sample performance.

Use the right weighting matrix

- Inverse clustered moment covariance matrix
  - This choice minimizes overall model error.
  - Takes into account moment covariances.
  - Bazdresch et al. (2018) note good finite sample performance.
- If at all possible, do this.

Do not treat the process of estimating the model mechanically!

- First, try to calibrate the model by hand.
- You will learn about which parts of the model are useful for understanding which parts of the data.
- You will learn a lot about the economics of the model.
- You will also learn a lot about identification.
- You will end up with a good starting value for your estimation.
- Only after you have done this should you start your estimation running.

You are going to have to minimize an objective function:

- You can't use a gradient based method unless you have a closed-form GMM or MLE problem
- Multi-start algorithms, combined with Nelder Meade work very poorly, except. . ..
- Tiktak algorithm (Arnoud, Guvenen and Kleineberg 2017).
- Use the simulated annealing (SA), particle swarm (PS), or differential evolution algorithm (DE) to avoid local minima
- DE and PS can be parallelized, but I have found that they can converge way too fast or not at all, depending on technical parameter settings. In technical terms, they're downright squirrelly.
- Use the same seed for the random-number generator each time you simulate data off the model. Do not mess up this step.

Software

- Do not use Matlab, R, Python, Octave, Gauss, or any other *interpreted* language.
- They are too slow!
- To estimate a model, you usually have to solve it ∼50,000 times.
- Use a compiled language: C, C++, Fortran, JIT compiled Python
- I hear Julia can be fast, but I have never tried, and I have heard that it is finicky.
- Learn how to exploit multiple processors, a graphics card, a supercomputer, ...

Remember that you are actually doing estimation

- Get the standard errors right.
- The actual data are usually not i.i.d.
- When estimating the covariance matrix for empirical moments, you must take into account
  - Heteroskedasticity
  - Time-series autocorrelation
  - Cross-sectional correlation
  - Serial correlation, including correlation across moments.
- You know what to do!

WHY GO STRUCTURAL? BECAUSE YOU GET TO DO IT ALL!

- Write down models, solve models numerically, gather data, do complicated econometrics, . . .
- Going structural may be right for you if . . .
  - . . .not much on your calendar for next few years
  - . . .emotionally robust

Dynamic models need empirical benchmarks!

- What features of the data should one use to estimate and evaluate dynamic models?
- What are the finite sample properties of the simulation estimators used for this purpose?
- How do you test the external validity of dynamic models?

These questions are interesting because dynamic models are inherently testable.

- Dynamic models provide an abundance of quantitative time series and cross sectional predictions.

- This richness allows researchers to discipline dynamic models more than static models.

The usual estimation method relies on arbitrary choices.

- Choose several interesting moments.
- Pick parameters to get model implied moments as close as possible to data moments.
- No guidance as to the choice of moments.

Our solution is to use data features that are common across models.

- The solution to any dynamic model is characterized by

Value function:	given the current state, what is the value of the firm?
Policy function:	given the current state, what are its optimal decisions?

- We use the *empirical* policy function (EPF) as a benchmark for estimation and testing.

Why are EPFs better than moments?

- For any two models that describe the same variables, . . .
- Using EPFs to estimate parameters provides a uniform method for comparing models.
- The moments come from simulating data from the policy functions, . . .
- So why not use the policy functions directly.

EPF-based estimation is better in finite samples.

- Monte Carlos show simulation estimators have low bias and mean squared error.
- EPF-based estimation has greater power to detect model misspecification.

To make our setting concrete, we consider a simple dynamic capital structure model...

- The firm maximizes distributions to shareholders.
- The firm chooses next period leverage and investment.

The firm uses capital to produce revenue.

constant returns profit function	$z = zK/K$
stochastic productivity shock	$\ln(z') = (1-\rho)\mu + \rho\ln(z) + \sigma\varepsilon', \; \varepsilon' \sim \mathcal{N}(0,1)$
investment	$i = K'/K - (1-\delta)$
investment adjustment costs	γi^2

The firm can issue risk-free, secured debt and hold cash.

leverage/cash	$p \lesseqgtr 0$
collateral constraint	$p' \leq \xi$
taxes	τ

The firm's sources and uses of funds identity defines payments to equity.

- sources minus uses of funds

$$e(p, p', i, z) = z(1-\tau) - i - \frac{\gamma i^2}{2} - p(1 + r(1-\tau)) + p'(1 - \delta + i)$$

- If $e > 0$, distributions
- If $e < 0$, proportional cost of equity issuance, λ

The Bellman equation is standard

$$\pi(p,z) = \max_{p',i} \left\{ e(p,p',i,z) + \beta \mathbb{E}\pi(p',z')(1-\delta+i) \right\}$$

The model has a solution characterized by a value and a policy function.

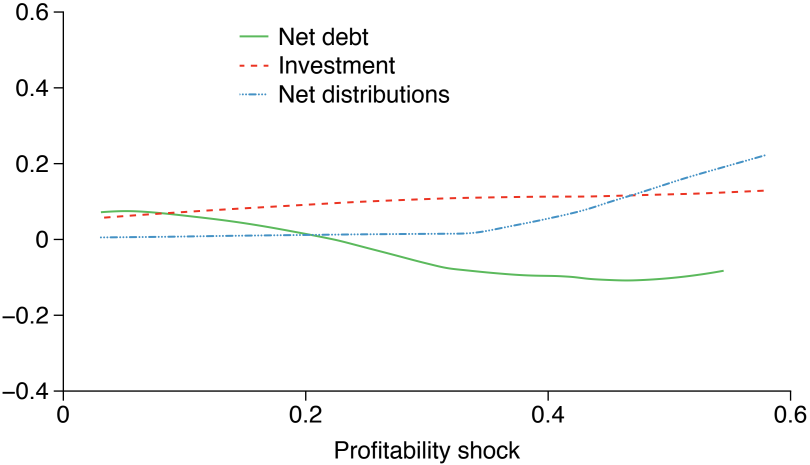
$$\text{Equity value} = V(\text{profitability, leverage})$$

$$\text{Investment and next period leverage} = G(\text{profitability, leverage})$$

control variables

state variables

Policy functions



Estimation is necessary to make the parameter values empirically relevant.

- SMM estimation.
- EPF-based estimation.

Identification of SMM requires arbitrary choices.

- You need to pick moments that vary with the underlying parameters.
- You should not cherry-pick moments.
- No guidance as to the choice of moments.

Identification of EPF-based estimation requires few arbitrary choices.

- Use any semiparametric regression technique to estimate the policy functions.
- Polynomial estimator.
- Capture level, slope, variability, curvature

We estimate the model using standard data.

- Compustat data from 1962 to 2012
- The model describes homogeneous firms.
- Remove time and firm fixed effects.
- But **levels** of variables are important.
- Add back sample mean.

Intuitive methods for estimating the parameters:

	Model Step		Data Step
SMM and EPF	Draw from profit shock		Collect data
	Compute optimal policies		Screen data
	Repeat		Remove fixed effects
SMM	Compute moments	$\Longleftrightarrow$ $\Downarrow$ parameters	Compute moments
EPF	Estimate policy functions	$\Longleftrightarrow$ $\Downarrow$ parameters	Estimate policy functions

The estimates are economically different.

Panel A: Parameters

Parameter	Moments-based	EPF-based
μ	-2.203	-2.172
	(0.014)	(0.002)
ρ	0.769	0.619
	(0.013)	(0.048)
σ	0.405	0.561
	(0.012)	(0.023)
δ	0.066	0.067
	(0.001)	(0.001)
γ	11.016	12.777
	(0.316)	(0.145)
ξ	0.444	0.493
	(0.022)	(1.088)
λ	0.236	0.059
	(0.021)	(0.945)
Overidentifying χ^2 *p*-value (d.f.)	0.000 (1)	0.000 (11)
Out-of-sample χ^2 *p*-value (d.f.)	0.000 (18)	0.000 (8)

The moments estimation does better at matching moments than the EPF estimation.

Panel B: Moments

	Data	Moments-based		EPF-based	
		Simulated	t-statistic	Simulated	t-statistic
Mean leverage	0.154	0.156	−6.187	0.215	−3.532
Variance leverage	0.021	0.020	18.722	0.045	−4.772
Mean investment	0.082	0.073	12.756	0.082	0.048
Variance investment	0.003	0.000	40.639	0.000	39.176
Mean distributions	0.006	0.015	−22.771	0.005	−0.212
Variance distributions	0.006	0.001	76.988	0.002	24.983
Mean profits	0.132	0.131	0.236	0.142	−9.729
Variance profits	0.007	0.006	12.321	0.010	−23.840

There are two interesting test stats in that table!

- The first is a standard t-stat on the moment conditions.
- To calculate the t-stats, you need to calculate the moment covariance matrix.

$$\text{var}(g(\boldsymbol{v}_{it},\theta)) = \frac{1}{nT}\left(1+\frac{1}{S}\right)(I - G(G'WG)^{-1}G'W)\hat{\Omega}(I - G(G'WG)^{-1}G'W).$$

We also develop an out-of-sample moment condition test.

- Suppose we have a vector of benchmarks, $m^*(\cdot)$, that are not used to estimate the parameter θ. We want to test the null hypothesis that

$$g^*\left(\boldsymbol{v}_{it}, \theta\right) = E\left(m^*\left(\boldsymbol{v}_{it}\right) - S^{-1}\sum_{s=1}^{S} m^*\left(\boldsymbol{v}_{it}^{s}\left(\theta\right)\right)\right) = 0.$$

- We need the variance of these moment conditions, which are given by:

$$\text{avar}(g^*\left(\boldsymbol{v}_{it}, \theta\right)) = E\left[\phi_g^* \phi_g^{*\prime}\right],$$

where ϕ_g^* is the influence function for g^*.

- The influence function for $g^*\left(\boldsymbol{v}_{it}, \theta\right)$ is given by:

$$\phi_g^* = \phi_m^* - \left(S^{-1}\sum_{s=1}^{S}\left(\partial m^*\left(\boldsymbol{v}_{it}^{s}\left(\theta\right)\right)/\partial\theta\right)\right)\phi_\theta.$$

Monte Carlo Design

1. Solve model at parameters estimates from polynomial EPF.
2. Simulate 1000 data sets
   - 20 years of data for 3,750 firms.
3. Estimate four parameters using EPF and SMM estimation:
   - depreciation rate δ.
   - equity issuance cost λ.
   - collateral value of capital ξ.
   - adjustment costs ψ.

Simulation estimators recover parameters very well.

	Moments-based		EPF-based	
Parameter	Identity	Clustered	Identity	Clustered
δ (depreciation rate)				
Average % bias	0.123	-0.006	-0.021	-0.001
RMSE %	0.608	0.047	0.059	0.010
$\Pr(t)$	0.605	0.367	0.309	0.348
λ (equity issuance cost)				
Average % bias	0.598	-0.165	1.793	-0.047
RMSE %	3.141	1.477	2.662	0.875
$\Pr(t)$	0.001	0.002	0.001	0.003
ξ (collateral parameter)				
Average % bias	-0.189	-0.299	-0.308	0.035
RMSE %	0.790	0.659	1.997	0.110
$\Pr(t)$	0.117	0.277	0.359	0.019
γ (investment adjustment cost)				
Average % bias	-0.239	0.027	0.052	0.007
RMSE %	1.273	0.106	0.127	0.022
$\Pr(t)$	0.313	0.135	0.115	0.100
Overidentification test rejection rate	0.558	0.048	0.825	0.083
External validity test rejection rate	0.985	0.843	0.668	0.079
Moment *t*-statistics:				
maximum rejection rate	0.317	0.022	0.354	0.024
median rejection rate	0.132	0.012	0.056	0.010
minimum rejection rate	0.000	0.005	0.000	0.005

Misspecification

- Estimate the same model, but introduce a misspecification.

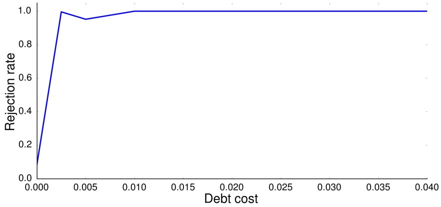
  Data model: Cost of debt issuance.

  Estimation model: No cost of debt issuance.

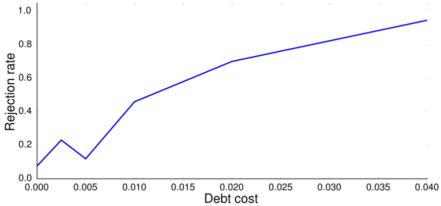
- Simulate 1000 data sets that follow the data model.
- Repeat Monte Carlo exercise using the estimation model.
- Count how many times model is rejected by standard χ^2 test.

EPF-based estimation test statistic power

Panel A: Overidentification test

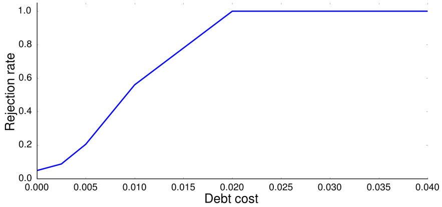


Panel B: Out-of-sample test

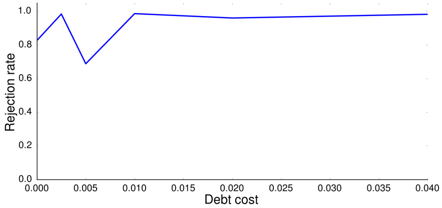


Moment-based estimation test statistic power

Panel A: Overidentification test



Panel B: Out-of-sample test



Andrews, I., Gentzkow, M.„ Shapiro, J.M., 2017. Measuring the sensitivity of parameter estimates to sample statistics. Quarterly Journal of Economics 132, 1553–1592.

Arnoud, A., Guvenen, F.„ Kleineberg, T., 2017. Benchmarking global optimization algorithms. Manuscript, Yale University.

Bazdresch, S., Kahn, R.J.„ Whited, T.M., 2018. Estimating and testing dynamic corporate finance models. Review of Financial Studies 31, 322–361.

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

Kahn, R.J.„ Whited, T.M., 2018. Identification is not causality, and vice-versa. Review of Corporate Finance Studies 7, 1–21.

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.