Estimation Project

Summer School on Structural Estimation in Corporate Finance

Overview

In this project you will perform a simple structural estimation of a workhorse model in corporate finance. The model features a firm that dynamically chooses its investment rate in the face of financing frictions. You are welcome to use any programming language you want, but the teaching assistants will mainly be familiar with Matlab, and solution code will only be provided in Matlab and Fortran.

Model

Your goal is to estimate the parameters α and δ in the following model. We are asking you to estimate only two parameters because we have limited time. If you are looking for a bigger challenge, you can try to estimate more than two parameters.

The setting is an infinite-horizon, discrete-time environment in which a manager acts on the part of shareholders to maximize the expected present value of their distributions. The firm uses capital in a decreasing-returns technology to generate operating income according to zK^{α}, $\alpha < 1$, in which α governs the degree of returns-to-scale, K is the capital stock, and z is a profitability shock. The profitability shock, z, is lognormally distributed and follows the process given by:

$$\ln(z') = \rho \ln(z) + \sigma \varepsilon', \qquad \varepsilon' \sim N(0,1). \tag{1}$$

Each period the firm chooses investment, I, which is defined by a standard capital stock accounting identity:

$$K' \equiv (1-\delta) K + I, \tag{2}$$

in which δ is the rate of capital depreciation. We normalize the price of capital goods to one. The firm's cash flow, $E^*(K, K', z)$, is its operating income minus its expenditure on investment, $K' - (1-\delta) K$:

$$E^* (K, K', z) = zK^\alpha - K' + (1-\delta) K. \tag{3}$$

Cash flows to shareholders, $E(K, K', z)$, are defined in terms of the firm's cash flows, $E^* (K, K', z)$. A positive firm cash flow is distributed to its stockholders, while a negative cash flow implies that the firm instead obtains funds from shareholders. In this case, the firm pays a linear cost, λ. Thus, shareholder cash flows are given by:

$$E^* \geq 0 \quad \Rightarrow \quad E = E^* \tag{4}$$

$$E^* < 0 \quad \Rightarrow \quad E = E^* (1+\lambda). \tag{5}$$

Having defined cash flows, we can now state the firm's problem:

$$\Pi (K, z) = \max_{K'} \{E (K, K', z) + \beta \mathbb{E} \Pi (K', z')\}, \tag{6}$$

subject to (2).

Day One (Monday)

Your goals for this day are to numerically solve the model and understand how it works. A deep understanding of the model's solution is critical for the rest of the project. You are encouraged to use the numerical techniques covered by Stephen Terry. Deliverables:

- Plot the value function against the state variables. What is the intuition behind the plot?
- Plot the firm's policy function against the state variables. What is the the intuition?
- How the do the two plots above change with the values of α and δ (the parameters you're trying to estimate)?

- Write a subroutine that takes the model's solution and parameters as inputs, then produces a simulated dataset as output. Make sure the random-number generator uses the same seed value every time the simulation is run, otherwise your estimation may not converge.

- To help you, we have put the file tauchen.m in the "Helpful code" folder. This code will help you compute expectations using the Tauchen (1986) method.

- If you finish early, start on day two's work.

Day Two (Tuesday)

Your goals for this day are to develop an identification strategy, choose features of the data to target in estimation, and write code for computing those features in the both the actual and simulated data. Remember that you're only trying to estimate α and δ. Set $\beta = 0.96$, $\rho = 0.75$, $\sigma = 0.30$, and $\lambda = 0.05$.

Deliverables:

- Which features of the data will you ask the model to fit? Those features can include means, variances, regression coefficients, correlations, etc. (I'll call these "moments," even though some of them are not technically moments.)

- Be able to explain which moment or moments are most important for identifying each model parameter. Create a table showing how each simulated moment changes as you perturb α and δ.

- Download the data you'll use to estimate the model. Data are in the file RealData.mat, or RealData.csv if you prefer not to work in Matlab. This file contains cleaned Compustat data on non-financial firms from 1971–2016. The variables we have chosen to put in the data file should give you a hint about the types of moments that will work well. You'll want to read RealData_Documentation.pdf, found in the Data folder. This short document describes the data and how the data variables map into the model variables.

- Write a Matlab routine for computing your vector of moments from the actual data and the simulated data. Hint: This code needs to work on both the actual and simulated data, so both those datasets need to be formatted the same.

- Start coding up the SMM routine. Write the code for finding the parameter vector that minimizes the econometric score (i.e., the distance between the actual and simulated moments).

- Hint: It is important to use a minimizer that will avoid local minima, so Matlab's fminsearch function will not work. We recommend using the simulated annealing minization algorithm. We have added the file anneal_1.m, which runs simulated annealing, in the "Helpful code" folder. That function is based on some code that Matlab put on its website 7+ years ago. Matlab may have a newer, fancier version online now. You will also find "how to use anneal_1.m" useful. Toni says she uses particle swarm now instead of simulated annealing.

- Hint: An important input to estimation is the SMM weighting matrix. On Wednesday you'll compute that weighting matrix correctly. For Tuesday's work, as a placeholder in your code, use any positive definite weight matrix for the weighting matrix.

- Hint: If one of your moments is an AR1 coefficient, use the Han and Philips method at end of Toni's slides from today.

Day Three (Wednesday)

Your goals for this day are to use influence functions to estimate the covariance matrix for the moments, compute the optimal weight matrix, code up the estimator, and debug it.

Deliverables:

- Use influence functions to estimate the covariance matrix for the moments you've chosen, clustering by firm.
- Compute the optimal SMM weight matrix.
- Finish coding the SMM routine if you haven't already.
- Code up the SMM standard errors and test of overidentifying restrictions.
- Start running and debugging the code that will choose the parameter values that minimize the score.

Day Four (Thursday)

Your goals for this day are to finish coding up the SMM estimator, run it, and also perform the test of overidentifying restrictions and counterfactual analyses.

Deliverables:

- Debug and catch up from Day 3 as needed.
- Compute your final parameter estimates and their standard errors. Do they make sense?
- (If you have extra time) How well does the model fit the data? Compare the simulated and empirical moments. Does the model fail the test of overidentifying restrictions?
- (If you have extra time) What would be an interesting counterfactual experiment to run? How would you do it?