Econ 388E - Fall 2021
Assignment 1: Introduction to Non-Linear Estimation Strategies and Simulation
Due October 1

In this problem set we will apply some of the techniques introduced in class. Questions will require the use of Matlab or a similar statistical/matrix programming package (e.g. Gauss, R, Julia, Python). The three questions increase in order of complexity, starting with a simple linear model, then a simple (invertible) non-linear model, and then a more complicated non-invertible, non-linear model that will require simulation. For references/practice on numeric optimization, you can play around with the matlab program "samp.m" on Canvas, look around on web for lecture notes (e.g. https://www.benfrederickson.com/numerical-optimization/), or maybe *Numeric Optimization* (Nocedal and Wright, 2006). To help you out so you know when you have debugged your programs :), here is a partial answer list of the coefficient estimates.

1) a, b, and c) $\widehat{\theta}_1 = 0.0055 \quad SE(\widehat{\theta}_1) = 0.0225$ (note: your standard errors could be different by, e.g. up to 5-10% due e.g. step sizes in the numeric derivative calculations.)

2)

a) $\widehat{\theta}_1 = 3.0659 \quad SE(\widehat{\theta}_1) = 0.1954$

b) $\widehat{\theta}_1 = 3.0574 \quad SE(\widehat{\theta}_1) = 0.1863$

3)

a) and b) it will depend a bit on which unobservable you simulate over in part (a) and what starting weight matrix you use in part (b), but you should get $\widehat{\theta}_1 \approx 1 \quad SE(\widehat{\theta}_1) \approx 0.009$

1) Linear Model

Consider the following linear model with two parameters:

$$y_i = \theta_1 + \theta_2 x_i + \epsilon_i$$

The file "data1.dat" contains data generated by this model. The first column is y_i, and the second column is x_i.

- - a) Estimate this model using OLS. Compute both parameter estimates of θ_1 and θ_2, as well as the standard errors of these estimates (according to the standard OLS formulas).
  - b) Under the assumption that the distribution of ϵ_i is i.i.d. (across observations) $N(0, \sigma^2)$, estimate this model using Maximum Likelihood. This will require use of an optimization procedure in Matlab (or R or Julia or Gauss), e.g. the Nelder-Mead simplex method (fminsearch in Matlab) or a derivative based routine (fminunc in Matlab). Note that there are now three parameters to estimate - θ_1, θ_2, and σ. Also note that the parameter σ is restricted to be greater than zero. While generic search procedures often do not directly support restrictions on parameter space, there is a simple way to enforce this restriction in your program. In the likelihood function procedure, simply set $\sigma = abs(\sigma)$ (a slightly more elegant alternative is to search over a tranformed parameter $\widetilde{\sigma} = \ln(\sigma)$ which has full support) Compute standard errors of your estimates. Recall that the asymptotic variance formula for an ML estimator can be expressed as

  $$Var(\theta) = \frac{1}{N} E\left[\frac{\partial \ln \ell_i}{\partial \theta}\frac{\partial \ln \ell_i}{\partial \theta'}\right]^{-1}$$

  which can be estimated with

  $$\widehat{Var(\theta)} = \frac{1}{N}\left[\frac{1}{N}\sum_{i=1}^{N}\frac{\partial \ln \ell_i(\widehat{\theta})}{\partial \theta}\frac{\partial \ln \ell_i(\widehat{\theta})}{\partial \theta'}\right]^{-1} = \left[\sum_{i=1}^{N}\frac{\partial \ln \ell_i(\widehat{\theta})}{\partial \theta}\frac{\partial \ln \ell_i(\widehat{\theta})}{\partial \theta'}\right]^{-1}$$

  where $\widehat{\theta}$ are your estimated parameters, and ℓ_i is the likelihood contribution of observation i, i.e. $\ell_i(\theta) = p(y_i|x_i, \theta)$. It is probably easiest to compute the derivatives of the individual likelihood contributions $\ell_i(\theta)$ numerically, i.e. to compute the first element of $\frac{\partial \ln \ell_i(\widehat{\theta})}{\partial \theta}$, use:

  $$\frac{\partial \ln \ell_i(\widehat{\theta})}{\partial \theta_1} \approx \frac{\ln \ell_i\begin{pmatrix} 1.001\widehat{\theta}_1 \\ \widehat{\theta}_2 \\ \widehat{\sigma} \end{pmatrix} - \ln \ell_i(\widehat{\theta})}{0.001\widehat{\theta}_1}$$

This is a "one-sided" numeric derivative. One could alternatively do a "two-sided" numeric derivative (or in some cases analytically solve for the derivatives). Check if your numeric derivatives are sensitive to the step size (in the above formula the step size is 0.1%).

– c) Under the assumption that $E[\epsilon_i|x_i] = 0$, estimate this model using GMM. Note that we are back to only two parameters now (σ will not be directly identified with only this mean independence assumption - think about and explain why this is the case). Use the moment condition:

$$G(\theta) = E\left[\epsilon_i(\theta) \otimes \begin{pmatrix} 1 \\ x_i \end{pmatrix}\right] = 0 \qquad \text{at } \theta = \theta^0$$

for estimation. Note that this moment condition is two dimensional. Since there are also two parameters, this model should be exactly identified, and you should be able to find a parameter vector that sets the sample analogue of this moment condition:

$$G_N(\theta) = \frac{1}{N}\sum_{i=1}^{N}\left(\epsilon_i(\theta) \otimes \begin{pmatrix} 1 \\ x_i \end{pmatrix}\right) = \frac{1}{N}\sum_{i=1}^{N} g_i(\theta)$$

exactly equal to zero. *Do not* simply solve out for $\widehat{\theta}$ (even though we know from class that this is possible for a linear model). Instead, use an optimization procedure in Matlab to find the $\widehat{\theta}$ that minimizes the quadratic form

$$G_N(\theta)' A G_N(\theta)$$

where A is a two by two weight matrix. Since the model is exactly identified, there should be a value of θ that sets $G_N(\theta)$ to exactly a zero vector. Hence, the choice of the weight matrix A should be irrelevant. However, it is still good practice to try to contruct an approximation to the optimal weight matrix. Recall that the optimal weight matrix is given by

$$\text{Optimal } A = Var\left(G_N(\theta_0)\right)^{-1}$$

Since you do not know θ_0, you can use some guess θ^{init} (this guess will not affect consistency) and use

$$\begin{aligned} A &= Var\left(G_N(\theta^{init})\right)^{-1} = \frac{1}{N}Var(g_i(\theta^{init})) \\ &\approx \frac{1}{N}\frac{1}{N}\sum_i\left(g_i(\theta^{init}) - \frac{1}{N}\sum_i g_i(\theta^{init})\right)\left(g_i(\theta^{init}) - \frac{1}{N}\sum_i g_i(\theta^{init})\right)' \end{aligned}$$

for your weight matrix. Then use a Matlab search procedure to compute your estimate, i.e.

$$\widehat{\theta} = \arg\min_{\theta} G_N(\theta)' A G_N(\theta)$$

Next, compute standard errors of your GMM estimates. Recall that the asymptotic variance formula for the GMM estimator is:

$$\left(\Gamma' A \Gamma\right)^{-1}\left(\Gamma' A V A \Gamma\right)\left(\Gamma' A \Gamma\right)^{-1}$$

where

$$\begin{aligned} \Gamma &= E\left[\frac{\partial G(\theta_0)}{\partial \theta'}\right] \approx \frac{\partial G_N(\widehat{\theta})}{\partial \theta'} \\ V &= Var(G_N(\theta_0)) = \frac{1}{N}Var(g_i(\theta_0)) = \frac{1}{N}E(g_i(\theta_0)g_i(\theta_0)') \approx \frac{1}{N}\frac{1}{N}\sum_i (g_i(\widehat{\theta})g_i(\widehat{\theta})') \end{aligned}$$

Note that we don't need to subtract out the means when estimating V, since by assumption, $Eg_i(\theta_0) = 0$. To estimate Γ, again use numeric derivatives, i.e. perturb $\widehat{\theta}$ slightly and recalculate $G_N(\widehat{\theta})$ (as in part b). Lastly, prove (and verify numerically), that since the model is exactly identified (number of moments = number of parameters), the variance formula actually reduces to

$$\Gamma^{-1} V \Gamma'^{-1}$$

2) A simple non-linear model:

Consider the following model

$$y_i = \exp(\theta_1 + x_i^{\theta_2} + \epsilon_i)$$

The file "data2.dat" contains some data generated by this model. Again, the first column is y_i, and the second column is x_i.

- – a) With the assumption that ϵ_i is distributed i.i.d. (across observations) $N(0, \sigma^2)$, estimate this model by Maximum Likelihood and compute standard errors. You may find it slightly more straightforward to derive the likelihood of the identical model:

  $$y_i = \exp(\theta_1 + x_i^{\theta_2} + \sigma\epsilon_i)$$

  where ϵ_i is distributed i.i.d. $N(0, 1)$. Again, you will be estimating parameters θ_1, θ_2, and σ. Given that this is not a linear model, it is possible that this likelihood function may be multi-modal, i.e. that there may be local maxima. *So, be sure to try many different sets of starting values in your estimation procedure.* If different starting values lead to different "maximum", the "maximum" with the highest likelihood would be your ML estimate.

  – b) Going back to the notation

  $$y_i = \exp(\theta_1 + x_i^{\theta_2} + \epsilon_i)$$

  and under the assumption that $E[\epsilon_i|x_i] = 0$, estimate this model using GMM (and compute standard errors). Again, use the moments

  $$G(\theta) = E\left[\epsilon_i(\theta) \otimes \begin{pmatrix} 1 \\ x_i \end{pmatrix}\right] = 0 \qquad \text{at } \theta = \theta^0$$

  Again, only two parameters will be identified by this moment (θ_1 and θ_2) (again, σ will not be directly identified with only this mean independence assumption), and since the moment is a 2 vector, the model should be exactly identified. Again, be careful with optimization (i.e., try lots of different starting values).

  – Which of a) and b) is robust to heteroskedasticity of ϵ_i with respect to x_i? Why?

  – Explain why this GMM estimator is not efficient (given the assumption that $E[\epsilon_i|x_i] = 0$)?

- 3) Cournot Duopoly - Suppose you have data generated by a number of Cournot duopoly markets (Cournot duopoly is when two firms simultaneously choose the quantities they will produce, and price is determined by the demand curve at that quantity). Suppose you know the inverse demand curve in each market i to be:

  $$p_i = 100 - q_i$$

  where $q_i = q_{i1} + q_{i2}$ is the total quantity produced by the two firms in market i. The two firms in each of the markets have different marginal costs, given by the functions:

  $$\begin{aligned} mc_{i1} &= \exp(\theta_1 + \theta_2 x_i^1 + \sigma\epsilon_i^1) \\ mc_{i2} &= \exp(\theta_1 + \theta_2 x_i^2 + \sigma\epsilon_i^2) \end{aligned}$$

  where x_i^1 and x_i^2 are the observable components of marginal cost (for firms 1 and 2 respectively in market i), and ϵ_i^1 and ϵ_i^2 are unobserved components. Note that θ_1, θ_2, and σ are the same for both firms. The file "data3.dat" contains data from these markets. The data is organized as follows - each row corresponds to a market, with three columns p_i, x_{i1}, and x_{i2} (note that we *do not observe* firm's individual quantities or market quantity).

  – a) Write down the profit maximizing first order conditional for the two firms. Simultaneously solve these first order conditions for equilibrium quanties, q_{i1} and q_{i2}, and the equation governing equilibrium price p_i. (Note: assume that both firms observe each others marginal costs, i.e. both firm 1 and firm 2 observe x_i^1, ϵ_i^1, x_i^2, and ϵ_i^2. In other words, firms observe the unobservables ϵ_i^1 and ϵ_i^2 - it is you, the researcher, who doesn't.)

– b) With the assumptions that ϵ_i^1 and ϵ_i^2 are distributed i.i.d. $N(0,1)$ (and independent of each other), estimate the parameters using maximum likelihood based on the likelihood of the observed p_i given the observed x_{i1}, and x_{i2} . Given that there are two unobservables and one dependent variable, note that the pricing model is not invertible. Thus, to construct the likelihood, you are going to have to "integrate out" *one* of the unobservables. You should simulate the resulting one-dimensional integral using 20 simulation draws. While it is best to use different simulation draws for the different observations, you should ensure that the draws are *held constant* for different θ's that you evaluate the likelihood at. One easy way to do this is to generate a 50 by 20 matrix of the underlying $N(0,1)$ random draws you will need ($N = 50$ markets by $S = 20$ simulation draws) at the beginning of your program, and then use that same matrix each time. So that we all get the same results, I will actually give you a set of simulation draws to use. They are in the file "drawsml.dat", which contains a 50 by 20 matrix of these draws. Note that the likelihood function involves taking a natural log (in addition to the log of the likelihood). For particular combinations of parameter values, observation number, and simulation draws, the quantity inside this natural log may be negative, which is obviously problematic. One way to address this (where z represents the potentially negative number), is to redefine $z = \max\{z, 0.0001\}$. This is a simple way to allow the program to run without crashing due to trying to take the natural log of a negative number.

– c) Why would it be challenging to estimate this model with only a mean independence assumption, e.g. $E\left[\epsilon_i^1|x_i\right] = E\left[\epsilon_i^2|x_i\right] = 0$?

– d) Again assuming that ϵ_i^1 and ϵ_i^2 are distributed i.i.d. $N(0,1)$, estimate this model using GMM (or more precisely MSM). Use the following "generic" moments:

$$E\left[\begin{array}{c} (p - E[p|x_i^1, x_i^2, \theta) \otimes \begin{pmatrix} 1 \\ x_i^1 \\ x_i^2 \end{pmatrix} \\ (p^2 - E[p^2|x_i^1, x_i^2, \theta) \end{array}\right] = 0 \qquad \text{at } \theta = \theta^0$$

where you simulate both of the inner expectations. In contrast to part (b) where you only integrated out one of the unobservables, here you will be simulating 2 dimensional integrals (you can again use 20 draws). The file "drawsgmm.dat" contains a 50 by 40 matrix of these draws (use the first 20 columns for draws on ϵ_i^1, the second 20 columns for ϵ_i^2). Note that the second moment (p squared) is intended to aid with identification of the parameter σ. Also note that this model is overidentified - there are 3 parameters (θ_1, θ_2, and σ) and four moments. As a result, you will need to choose a weight matrix A. Recall that the "optimal" A is $A = V^{-1} = Var(G_N(\theta^0))^{-1}$. Initially set $A^{init} = V^{-1} = Var(G_N(\theta^{init}))^{-1}$, where θ^{init} is an initial guess at θ (perhaps use some crude assumption like $p = mc$, in which case you could use OLS and regress $\ln(p)$'s on a constant and x_i's to get some (incorrect "estimate" of θ_1, θ_2, and σ) . Given that θ^{init} may be far away from θ^0, you should probably subtract off means in computing V, i.e.

$$V = Var(G_N(\theta^{init})) \approx \frac{1}{N}\frac{1}{N}\sum_i (g_i(\theta^{init}) - \frac{1}{N}\sum_i g_i(\theta^{init}))(g_i(\theta^{init}) - \frac{1}{N}\sum_i g_i(\theta^{init}))'$$

(near θ^0 these means should be zero). Estimate the model using this weight matrix, resulting in an initial estimate $\widehat{\theta}$. Next, do a second stage - i.e. recompute $A = V^{-1} = Var(G_N(\widehat{\theta}))^{-1}$ and use this A as the weight matrix in your optimization procedure. Using the formulas in question 1 (though now the variance formula does not simplify because the model is overidentified) compute standard errors after *both* stages. Do they decrease after the second stage?

– e) Try estimating the model with only the first three moments, and recalculate standard errors. What is the most notable change to the standard errors of the estimated parameters? Can you explain why it might be important to include the 4th moment condition above (i.e. the one with p^2)? I'm not looking for a formal argument here, just some loose intuition.

– f) Suppose demand is now given by:

$$p_i = 100 - q_i + \phi\eta_i$$

where $\eta_i \sim N(0,1)$ is an unobservable and ϕ is an additional parameter. Describe (briefly) what would change in estimating this model (both ML and GMM) (You do not need to estimate this model!). How does this model generalize the original model? (Note: for purposes of the firms FOCs you can assume that the firms do not observe η_i and simply assume $\eta_i = 0$) .

– g) Suppose costs are instead given by:

$$\begin{aligned} mc_{i1} &= \exp(\theta_1 + \theta_2 x_i^1 + \sigma\epsilon_i^1 + \rho\alpha_i) \\ mc_{i2} &= \exp(\theta_1 + \theta_2 x_i^2 + \sigma\epsilon_i^2 + \rho\alpha_i) \end{aligned}$$

where $\alpha_i \sim N(0,1)$ is an econometric unobservable (but observed by the firms) and ρ is an additional parameter. Again, briefly describe how you estimation procedure would change (again, - you do not need to actually estimate the model!). How does this model generalize the original model (e.g. what does it mean when ρ is large vs close to zero?)

– h) Suppose that in addition to the data on p_i, x_i^1, and x_i^2, you additionally observed data on firm quantities q_i^1 and q_i^2 in each market. In model f), are there observable patterns in this data might help pin down the new parameters ϕ and σ? In other words, how would the data look in a world where ϕ is large and σ is small, versus a world where ϕ is small and σ is large? (Hint: what would data on p_i, q_i^1, and q_i^2 look like in a set of markets with similar x's) Would your answer change if you only observed $q_i = q_i^1 + q_i^2$? What if you were back to only observing p_i, x_i^1, and x_i^2, i.e. no quantities?

– i) In the model in g), again assuming that you observe q_i^1 and q_i^2, are there observable patterns in the data might help pin down the new parameters ρ and σ? In other words, how would the data look in a world where ρ is large and σ is small, versus a world where ρ is small and σ is large? How would your answer change if you only observed $q_i = q_i^1 + q_i^2$?