Problem Set # 1: Basic programming

Daniel Herrera-Araujo
Course: Empirical IO

August 3, 2022

This first problem set is meant to introduce you advanced functions in MATLAB. It works over the introductory course of Matlab done by Cedric and Ines. You will implement an OLS estimator, maximum likehood estimation (MLE) and Simulated MLE.[1]

Problem 1- Estimating OLS and MLE using automobile data. We will start import the dataset "imports-85.csv" into using in-house Matlab import commands and then save it to with the name "imports.mat".[2]

1 Create X, a $N \times 6$ matrix using the following variables as columns: a constant equal to 1, length, curb weight, engine size, horsepower, city-mpg.[3] Suppose that we only have access to a pricing variable with reduced information. Construct a dichotomous pricing variable equal to one if the price is above the median, and zero otherwise. You will need to make sure that the price and covariates have the same dimension.

2 Estimate the model as indicated below:

   a Estimate the model using the analytical formula of the Ordinary Least Square estimator (i.e, $(X'X)^{-1}X'y$ with $X = [1, X_1]$ a $N \times 6$ matrix) and calculate the standard errors.

   b Estimate the model minimizing the sum of squared residuals, (i.e., $\sum_{i=1}^{N}(y_i - \beta_0 - \beta_1 X_{1i})^2$ with an optimization code that you need to write.

   c Obtain the standard errors for 2a and 2b. (tip: use the Hessian).

3 Please estimate the model, and its standard errors, using MLE assuming that the error term follows a normal distribution.

[1] This problem set is done in collaboration with Paul-Emile Bernard.

[2] For more information about the dataset follow the link, https://www.rdocumentation.org/packages/randomForest/versions/4.7-1.1/topics/imports85 [last visited on August 2022].

[3] Hint: Use the grp2idx command to transform categorical variables into numerical ones. Remove NAN with rmmissing() command.

4 Overlay a histogram of the error terms and the pdf of a normal distribution using as standard deviation the one that is estimated using the MLE results. You will need to first generate the error term (use the coefficients from the MLE. To generate the histogram use the histogram() command. Use the fitdist() command to fit a normal distribution to the error term. Please comment the results. Why is linear probability model not suited?

5 Estimate a logit model using fminsearch. You can proceed by creating a function that calculates the cumulative for a logitic model distribution function (or just exploit the closed form formula of the cumulative function of a logit. Always remember that fminsearch minimizes, so you need to put a negative sign to the output of the fminsearh.

Problem 2- Estimating simple models by Simulated Maximum likelihood (SML). Suppose that the outcome variable y_{ij} for $j = \{1, 2, 3\}$ can take one of two values: 0,1. There are $i = 1, \ldots, N$ observations, with N = 10000. Parameters θ govern the distribution of y_{ij}, $P[y_{ij}|\theta]$. Define $N_1 = \sum_i d_{i1}$, where $d_{i1} = 1$ if y_{i1} is equal to 1 and 0 otherwise. Similarly, N_2 and N_3 denote the number of y_{i2} and y_{i3} that equal 1, respectively. Note that by definition, $N = N_1 + N_2 + N_3$. For a given value of θ, the likelihood function is given by:

$$L(\mathbf{y}, \theta) = \prod_i \prod_j P[y_{ij} = 1|\theta]^{d_{ij}}.$$

Let the log-likelihood be $l(\mathbf{y}, \theta) = \ln L(\mathbf{y}, \theta)$. Then the maximum likelihood estimator of θ is given by:

$$\theta_{ML} = \arg\max_{\theta} l(\mathbf{y}, \theta)$$

We will carry out the estimation numerically using simulations.

0. Let $\theta = [\theta_1, \theta_2, \theta_3]$ denote a 3×1 vector. Next, let $X = [X_1, X_2, X_3]$ where X_j for $j = (1, 2, 3)$ is a $N \times 1$ vector. Then, let $\epsilon = [\epsilon_1, \epsilon_2, \epsilon_3]$ where ϵ_j for $j = (1, 2, 3)$ follows an extreme value distribution. Finally, let $U_{ij} = X_j * \theta_j + \epsilon_{ij}$. Define $P[y_{ij} = 1|\theta] = (1/N) \sum_i 1(U_{ij} \geq \max(U_{i1}, U_{i2}, U_{i3}))$

1. First start creating a fake dataset using the above mentioned DGP. To do so, please follow the following steps:

   a. Generate X a $N \times 3$ matrix from a standard normal distribution with mean **2** and variance $\Sigma = diag(1)$. Similarly, generate ϵ a $N \times 3$ matrix from an extreme value distribution. The three vectors of ϵ should be independent from each other.

   b. Set $\theta_1 = -1$, $\theta_2 = 1$ and $\theta_3 = 0$. Then, generate the outcome y_{ij} by setting $y_{ij} = 1$ if $U_{ij} \geq \max(U_{i1}, U_{i2}, U_{i3})$ for each $i = (1, 2, 3)$.

   c. Generate the counters N_1, N_2 and N_3.

   d. Save the fake dataset.

2 Compute the SML estimator of θ. To do so, follow the next steps

a. Generate S=100 matrices of size $N \times 3$ with random draws from an extreme value distribution. For simplicity, label as ϵ_{s1} the first $N \times 3$ matrix, ϵ_{s2} the second, and so on.

b. Generate three vectors containing $e_1 = 1, \ldots, E_1$, $e_2 = 1, \ldots, E_2$ and $e_3 = 1, \ldots, E_3$ possible values for parameters θ_1, θ_2 and θ_3 assuming that $\theta_1 \in [-1.5, -0.01]$, $\theta_2 \in [0.01, 1.5]$ and $\theta_3 = 0$. (Hint: exploit Matlab command from:sizeofstep:to).

c. - Take 1 draw of $[\theta_1^{e_1}, \theta_2^{e_2}, 0]$

c1. - For each $\epsilon_{s1} \ldots \epsilon_{s100}$ generate U_s with its corresponding outcome y_{ijs}.

c2. - Generate counters S_1^s, S_2^s and S_3^s

c3. - Compute and save $P[y_{ijs} = 1|S] = S_j^s/N$ for $j = \{1, 2, 3\}$.

c4. Average the simulated probabilities $PS_j = \sum_{s=1}^{100} P[y_{ijs} = 1|S]/100$

c5. Compute and save the natural logarithm of the simulated likelihood function:

$$PS_1^{N_1} PS_2^{SN_2} PS_3^{SN_3}.$$

d. Repeat c1-c5 for all other $[\theta_1^{e_1}, \theta_2^{e_2}, 0]$.

f. Find the maximum and the maximizer of the log-simulated likelihood function over all the possible values of $[\theta_1, \theta_2, 0]$ using the in-built function "max".

3 Re-estimate the SML estimator of θ using the built-in function "fminsearch". Note that "fminsearch" performs minimizations, while you need to maximize the function. Use many different starting values and select the lowest l value. [Hint: you need to create a file function using as basis the log-likelihood. Remember that fminsearch minimizes, so you need to put a negative sign to the $l(\mathbf{y}, \theta)$.]