Assignment 2

Hema Yoganarasimhan

Course Number: MKTG 584 A
Course Name: Dynamic Structural Models

1 Overview

This assignment follows up on your Assignment 1. Assume the same optimal stopping problem (and model specifications) as specified in Assignment 1.

The goal of this assignment is to estimate the replacement costs and maintainence costs for a given dataset using the Nested Fixed Point Algorithm.

2 Inputs

Your code should take the following inputs:

- Data in following format (in columns) – BusNo., Timeperiod, DecisionNo. (0 if the line refers to continuation, 1 if it refers to replacement), Mileage, Chosen (0 if this line was chosen, 1 otherwise). Note that this is the same format as the data you generated in Assignment 1, except it doesn't have the column for the choice-specific value function.
- Discount factor β

3 Outputs

For a given dataset and discount factor, the code should generate the following outputs:

- Parameters $\{\theta_1, \theta_2\}$ and the standard errors for each of them.

4 Outline of the Code

There are two parts to this code – the outer loop and the inner loop. The outer loop consists of maximum likelihood estimation to update the θs for a given set of value functions. The inner loop consists of value function iteration (to update the value functions) for a given set of θs. The program converges when the θ values have converged. Set the convergence criterion to be sufficiently low.

You already have the inner loop from Assignment 1. So the goal is to super-impose an outer loop. There are two options to do this – 1) you could code a ML estimator in the language that you wrote the inner loop in, or 2) you could use a canned estimator in a different language. The advantage of the latter approach is that you will be using a ML estimator without actually coding one. This is useful for two reasons. First, it ensures correctness. Second, a good ML estimator (such as asclogit from Stata; described below) is likely to have analytical first and second derivatives coded, which will increase the speed of your algorithm and the numerical precision of your estimates.

If you are not going to use a ready-made ML estimator for the outer-loop, you would need to write a ML estimator in the programming language you are working with. This will be a more involved process, depending on your familiarity with coding MLs. However, it will be a good learning experience, and will help you prepare for the future assignments.

Finally, because your value function is "estimated" instead of data, your standard errors of the parameter estimates are likely to be biased downwards. To address this problem, use bootstrap as follows:

1. Draw a sample of buses with replacement (the sampling is done at the bus-level). This sample should be large enough (about the size of the data).
2. Now treat this sample as your data and run your model on it. You will get a parameter estimate for θ_1 and θ_2. Store these estimates and ignore the accompanying standard errors.
3. Repeat steps 1 and 2 for a large number of samples, say 250.
4. The mean of the estimates for the 250 samples is the bootstrapped parameter estimate. And the bootstrapped standard error is the standard error of the parameter estimates for the 250 samples.

5 Some suggestions

- To simplify the analysis, assume that you know the state transition probabilities, $i.e.$, don't estimate them from data.
- Ensure that your code can accept the data in .txt format.
- Numerical precision can be an issue when working with exponentiated variables and your code may run into overflow problems or have difficulty converging. To avoid this problem, employ these fixes:

$$log(e^x + e^y) = max(x, y) + log(e^{x-max(x,y)} + e^{y-max(x,y)}) \quad (1)$$

$$\frac{e^x}{e^x + e^y} = \frac{e^{x-max(x,y)}}{e^{x-max(x,y)} + e^{y-max(x,y)}} \quad (2)$$

- If you are going to use Stata's asclogit command to run the outer-loop of ML, read the notes for asclogit posted under Class 1 files. Then write a Stata .do file which pulls the data from the last iteration, does the ML estimation, and spits out the new parameter values at this iteration. You can call a Stata .do file from Python, C, R, (and vice-versa) using simple one line commands.

- If you use Stata's asclogit for ML estimation, the following hints will be useful in storing the parameters for the next iteration of the inner-loop:
  - Stata stores all the output parameters after a given command in a matrix referred to as $e(b)$.
  - To automate your program, you would need to get the parameters stored under $e(b)$ and use them calculate the value functions at each iteration of your outer-loop. There is a simple user-written module called mat2txt that pulls and stores the $e(b)$ matrix in a text file. Download it in Stata using the command – ssc install mat2txt. Read the associated help files to understand how to use this command.
- Set the convergence critierion to be sufficiently low (at least 10^{-5}) for both the convergence of θs and the value function iteration.

6 Evaluation

For the submission, you need to provide the following:

- Your complete code with comments. I will evaluate the correctness of your code by giving you a .txt dataset in the format described above and a discount factor. Your code should produce the correct parameters for the data.
- A pdf document with the following results:
  - Run your model on the dataset I provided with this assignment and present the following – a) the parameter estimates and standard errors without bootstrap (run on the full data), and b) the bootstrapped parameter estimates and standard errors with at least 250 replications.
  - The probability distribution of parameters from the bootstrap process.