Semester Project - Weekly Report

Alexia Paratte

March 2024

Week 1 (29/07/24 - 07/03/24)

Goals

After having the first meeting on the 29/07/2024, these are the goals set for the first week:

- Basic organisation of the *Github*
- Choose an MNL on the *Apollo* website and estimate it
- Translate to *Biogeme*
- Insure the two models are equivalent same of threads (should be 8)
- see CPU and RAM used in both cases
- Look at the second derivative (Hessian) in `biogeme.toml`, which can go from 100 % to 0. Try to change the parameter: use 100%, 0% and some in between
- Change the method of estimating by modifying the file `biogeme.toml`. Need to change `optimization_algorithm` from `simple_bounds` (default) to `simple_bounds_BFGS`. Need to look at the *Biogeme* documentation to see exactly which one to choose

Progress

Intro

We choose the most basic MNL model on the *Apollo* website, which is the MNL_RP. It is a simple MNL model on mode choice RP data. An code in R is available, thus we run it using *RStudio* and the *Apollo* Librairy. You might see the code here. I have then translated it in *Python*, using the *Biogeme* library. The code can be found here. We have to make sure that the number of threads is the same in both cases (it is set to 8).

Data description

Our first resource is a synthetic dataset looking at mode choice for 500 travellers. For each individual, the data contains two revealed preference (RP) inter-city trips, where the possible

modes were **car**, **bus**, **air** and **rail**, and where each individual has at least two of these four modes available to them. The journey options are described on the basis of access time (except for car), travel time and cost, with times in minutes, and costs in $. For each individual, the dataset also contains information on gender, whether the journey was a business trip or not, and the individual's income. We remove the stated preference (SP) in this case.

Choice Model, Utility functions and parameters estimated
The choice model is a Multinomial Logit Model (MNL) with the following utility functions:

$$
\begin{aligned}
V_{\text{car}} &= \text{asc\_car} + \beta_{\text{tt\_car}} \cdot \text{time\_car} + \beta_{\text{cost}} \cdot \text{cost\_car} \\
V_{\text{bus}} &= \text{asc\_bus} + \beta_{\text{tt\_bus}} \cdot \text{time\_bus} + \beta_{\text{access}} \cdot \text{access\_bus} + \beta_{\text{cost}} \cdot \text{cost\_bus} \\
V_{\text{air}} &= \text{asc\_air} + \beta_{\text{tt\_air}} \cdot \text{time\_air} + \beta_{\text{access}} \cdot \text{access\_air} + \beta_{\text{cost}} \cdot \text{cost\_air} \\
V_{\text{rail}} &= \text{asc\_rail} + \beta_{\text{tt\_rail}} \cdot \text{time\_rail} + \beta_{\text{access}} \cdot \text{access\_rail} + \beta_{\text{cost}} \cdot \text{cost\_rail}
\end{aligned}
$$

We fix the asc_{car} parameter to 0, the other parameters (9 in total) need to be estimated. After running both code using *Apollo* in *RStudio* and *Biogeme* in *Python*, these are the parameters estimations:

Parameter	**Value in Apollo**	**Value in Biogeme**
asc_{car}	0	0
asc_{bus}	0.475	-0.290
asc_{air}	1.630	-0.335
asc_{rail}	0.945	0.145
$\beta_{\text{tt\_car}}$	-0.004	0.008
$\beta_{\text{tt\_bus}}$	-0.009	0.003
$\beta_{\text{tt\_air}}$	-0.021	0.020
$\beta_{\text{tt\_rail}}$	-0.011	0.013
β_{access}	-0.011	0.021
β_{cost}	-0.034	-0.002

Table 1: Parameter values MNL_RP Model in both *Apollo* and *Biogeme* librairies.

CPU and RAM monitoring
In order to monitor the CPU and the RAM when the programs are running, I have coded a function `monitor_system()` that takes the required duration of running into argument, and will compute the CPU used (in %) and the RAM used (in %) every 0.5 seconds. This could help us compare how the CPU and the RAM is used in both codes. For now, we get the results in Table 2. The time to run the code are 1.521 and 1.524 for *Apollo* and *Biogeme* respectively. I think the model is not complex enough to see a real difference for now.

Time(s)	CPU(%) Apollo/Biogeme	RAM(%) Apollo/Biogeme
0	2.7 / 3.7	54.3 / 47.0
0.5	2.3 / 7.8	54.4 / 47.0
1	3.1 / 3.8	54.1 / 47.0
1.5	2.0 / 6.6	54.1 / 47.0
2	9.5 / 8.7	54.1 / 47.2
2.5	9.7 / 3.9	54.4 / 47.4
3	10.2 / 13.6	54.3 / 47.6
3.5	7.8 / 4.6	54.4 / 47.6
4	3.4 / 1.1	54.4 / 47.6
4.5	2.3 / 4.1	54.4 / 47.6

Table 2

Modifying biogeme.toml
We want to try modifying two parameters in `biogeme.toml`, as to see how this changes the estimation of our model:

1. `second_derivatives`: proportion (between 0 and 1) of iterations when the analytical Hessian is calculated

2. `optimization_algorithm`: optimization algorithm to be used for estimation. The basic value is `simple_bounds`. Want to try out either `simple_bounds_newton` or `simple_bounds_BFGS`.

`optimization_algorithm`: the optimization algorithm in *Apollo* is using a Newton optimization algorithm (need to recheck where it was written in the documentation), and we want to impose the same for the *Biogeme* Librairy.

Questioning/Remarks

1. *Should we clear the memory at the beginning of the code in Python ?* `gc.collect()` *function ?* Need to restart the Kernel every time, `gc.collect()` does not seem to work

2. *Want to find a way to launch both code when running the* `monitor_system()` *function. This would help be more precise as to when exactly is the code running affecting the CPU and RAM (should be able to launch the code after 5 seconds of running the* `monitor_system()` *function for example).*

3. *Need to find a way to do a mean of the running time of each code (multiple iterations).*

4. *what should we measure when changing the* `second_derivatives` *and* `optimization_algorithm` *parameters ?*