EPFL
ENAC TRANSP-OR
Prof. M. Bierlaire

Mathematical Modeling of Behavior
Fall 2020

Netherlands Mode Choice

The objective of labs 7 and 8 is to become familiar with the use of choice models to predict aggregated market shares in hypothetical situations, to compute relevant indicators for policy analysis and to forecast the prices that maximize the expected revenue. To this end, we will use the simulation features of Biogeme.

The idea is to predict the market shares with a method called *sample enumeration*. In order to use the Netherlands dataset for aggregation and prediction, we have to restrict it to contain only the revealed preference (RP) data (i.e., the actual choices). From now on, when we refer to the Netherlands dataset, we refer to its restricted version containing RP data only. This dataset is called `netherlandsRP.dat` and is provided.

In this exercise, we consider a choice model that has been estimated for the Netherlands case study (see `Netherlands_Base_Model.ipynb`). The deterministic terms of the utility function are defined as follows:

$$\begin{aligned} V_{CAR} &= \mathrm{ASC}_{\mathrm{CAR}} + \beta_{\mathrm{Cost\_Age1}} \cdot \mathrm{Cost}_{\mathrm{CAR}} \cdot \mathrm{Age}_1 + \beta_{\mathrm{Cost\_Age2}} \cdot \mathrm{Cost}_{\mathrm{CAR}} \cdot \mathrm{Age}_2 \\ &\quad + \beta_{\mathrm{Time\_CAR}} \cdot \mathrm{TravelTime}_{\mathrm{CAR}}, \\ V_{RAIL} &= \beta_{\mathrm{Cost\_Age1}} \cdot \mathrm{Cost}_{\mathrm{RAIL}} \cdot \mathrm{Age}_1 + \beta_{\mathrm{Cost\_Age2}} \cdot \mathrm{Cost}_{\mathrm{RAIL}} \cdot \mathrm{Age}_2 \\ &\quad + \beta_{\mathrm{Time\_RAIL}} \cdot \mathrm{TravelTime}_{\mathrm{RAIL}} + \beta_{\mathrm{Female}} \cdot \mathrm{Female} + \beta_{\mathrm{ArrivalTime}} \cdot \mathrm{ArrivalTime}, \end{aligned}$$

where $\mathrm{Cost}_{\mathrm{CAR}}$ and $\mathrm{Cost}_{\mathrm{RAIL}}$ are the cost of car and rail in euros, respectively. $\mathrm{TravelTime}_{\mathrm{CAR}}$ and $\mathrm{TravelTime}_{\mathrm{RAIL}}$ are the total travel time of car and rail in hours, respectively. Age_1 is 1 if the individual is 40 years or younger, Age_2 is 1 if the individual is 41 years or older (i.e., $\mathrm{Age}_2 = 1 - \mathrm{Age}_1$), Female is 1 if the reported gender is female, and ArrivalTime is 1 if the individual must arrive at the destination by a given time.

1 Aggregation

We want to perform an aggregation to predict market shares, assuming that the procedure to collect the sample is *stratified random sampling*. Thus, it is necessary to associate a weight with each group or stratum. In our case, and given the available data and the relevant socioeconomic characteristics included in the choice model, we consider 4 strata:

1. Male and Age_1;
2. Male and Age_2;

3. Female and Age_1;

4. Female and Age_2.

In order to compute the weight associated with each group g, we need the total number of individuals in the population (N) and the number of individuals in the population belonging to each group (N_g). Even if the population should ideally contain only the individuals in the Netherlands commuting between Nijmegen and the Randstad in 1987 (scope of the case study), due to the unavailability of the data for this population, we consider the above mentioned strata for the whole country in 2011 as an instance.

	Age_1	Age_2
Male	4,092,390	4,151,092
Female	3,984,028	4,428,289

Table 1: Dutch Census 2011

Table 1 shows the size of the 4 groups in the whole population. We need to identify the size of these four groups S_1, S_2, S_3 and S_4 within the sample of dataset you have. This can be done using Pandas dataframes, or using a statistical software like Excel or R. We should obtain the following results:

1. $S_1 = 89$,
2. $S_2 = 37$,
3. $S_3 = 65$,
4. $S_4 = 37$.

Now, we need to compute the weight associated with each individual. The weight of group g is given by:

$$\omega_g = \frac{N_g}{N} \cdot \frac{S}{S_g}, \tag{1}$$

where $N = \sum_{g=1}^{4} N_g$ and $S = \sum_{g=1}^{4} S_g$. We should obtain the following weights:

1. $\omega_1 = 0.63$,
2. $\omega_2 = 1.54$,
3. $\omega_3 = 0.84$,
4. $\omega_4 = 1.64$.

Finally, we need to associate a weight with each individual. As each individual n belongs to exactly one stratum g, the individual weight is defined as:

$$\omega_n = \sum_{g=1}^{4} \delta_{ng} \omega_g, \tag{2}$$

where $\delta_{ng} = 1$ if individual n belongs to stratum g and $\delta_{ng} = 0$ otherwise.

Once the individual weights are computed, we can add them as a new column `Weights` in the dataset `netherlandsRP.dat`. This is the dataset that we will use from now on.

NB: Biogeme only reads tab-separated files, so make sure that your new dataset is saved in the correct format.

We need to guarantee that the sum of the individual weights is equal to the number of observations in the sample ($\sum_{n=1}^{S} \omega_n = S$). To do this, we normalize the weights directly in Biogeme as follows:

```
sumWeights = database.data['Weights'].sum()
S = database.getSampleSize()
sampleNormalizedWeight = Weights * S / sumWeights
```

2 Simulation file

Up to this point, we have added the column containing the weights to the Netherlands dataset. We will now perform the simulation task following the instructions provided in the Biogeme documentation (`https://biogeme.epfl.ch/documents.html > Calculating indicators with PandasBiogeme > Market shares and revenues`). The simulation should be performed in a new file that we will call `Netherlands_Base_Simul.ipynb`.

1. *Compute the predicted market shares for car and rail with stratified random sampling.* Follow instructions 1-2 from the documentation, using as base the model specification provided in `Netherlands_Base_Model.ipynb`. In order to get the predicted market shares, we need to define a dictionary `simulate` that will be passed as an argument to the model:

   ```
   prob_car = logit(V, av, 0)
   prob_rail = logit(V, av, 1)

   simulate = {
   'Prob. car': prob_car,
   'Prob. rail': prob_rail,
   'Weighted prob. car': sampleNormalizedWeight * prob_car,
   'Weighted prob. rail': sampleNormalizedWeight * prob_rail
   }
   ```

   These instructions can be added after you have defined your utility functions and availabilities. The function `logit` needs to be imported from `biogeme.models`.

   Follow instructions 5-7 from the documentation to perform the simulation. You can then retrieve the results to compute the aggregated market shares:

```
marketShare_car = 100 * simulatedValues['Weighted prob. car'].mean()
marketShare_rail = 100 * simulatedValues['Weighted prob. rail'].mean()
```

You should get the following results:

- car: 63.95%
- rail: 36.05%.

2. *Compare the predicted market shares with the actual choices. More precisely calculate the following shares:*

   - *share of users choosing car with a higher (unweighted) probability for rail, and*
   - *share of users choosing rail with a higher (unweighted) probability for car.*

   Try to find the possible causes.

   You should obtain:

   - share of users choosing car with a higher probability for rail: 9.21%
   - share of users choosing rail with a higher probability for car: 14.91%.

---

jp/ th / an / no / mpp