Data description

Antonin Danalet * Loïc Tinguely * Matthieu de Lapparent *
Michel Bierlaire *

29th October 2017

Transport and Mobility Laboratory
Ecole Polytechnique Fédérale de Lausanne
transp-or.epfl.ch

Abstract

This document describes the softwares, data, models and raw results associated to Danalet et al. (2016) and Danalet (2015).

Keywords

location choice; panel data; pedestrians; dynamic model; initial conditions problem

Conditions of use

You are free to share, use and remix the data associated to this document, but not for commercial purposes. When you share, use and remix the data, please mention our paper:

> Danalet, A., Tinguely, L., Lapparent, M. d. and Bierlaire, M. (2016). Location choice with longitudinal WiFi data, Journal of Choice Modelling 18: 1–17. URL: https://dx.doi.org/10.1016/j.jocm.2016.04.003

If you are interested in using these data for commercial purposes, please contact Antonin Danalet, `antonin.danalet@alumni.epfl.ch`

*ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE (EPFL), School of Architecture, Civil and Environmental Engineering (ENAC), Transport and Mobility Laboratory (TRANSP-OR), antonin.danalet@alumni.epfl.ch, loic.tinguely@alumni.epfl.ch, matthieu.delapparent@epfl.ch, michel.bierlaire@epfl.ch

Contents

1 Softwares

Two softwares have been used to generate results:

Pythonbiogeme Biogeme (Bierlaire; 2003) is an open source freeware designed for the estimation of discrete choice models. We specifically used pythonbiogeme (Bierlaire and Fetiarison; 2009), version 2.3. More information on `http://biogeme.epfl.ch/`. For help, consult the users' group: `https://groups.google.com/d/forum/biogeme`.

OpenOffice In particular the spreadsheet component, Calc. More information on `http://openoffice.org/`.

2 Data

`/data/`

We use WiFi traces to detect sequences of activity episodes. WiFi traces are merged with map information (localization of points of interest), attractivity (aggregate measures of occupancy, e.g., from point-of-sale data) and time constraints (e.g., shop opening or class schedules), as described in Danalet et al. (2014), with $L = 1$. This Bayesian approach merges data, detects stops and give semantics to the WiFi traces. The raw data are available in Danalet, Antonin (2015). Note that we have access to some socio-economic attributes in this dataset. We associated MAC addresses to usernames using the radius server, and then usernames to employee or class attributes using LDAP. Finally, usernames and MAC addresses have been deleted (Danalet et al.; 2014).

For exploring data, use any text editor or, more conveniently, a spreadsheet software, e.g., Calc in OpenOffice. All data files are text files with a first line of headers and subsequent lines of observations, with columns separated by tabs.

2.1 Dataset for estimation

`/data/1Estimation/Dataset.dat`

The full dataset, `Dataset.dat`, is used for the estimation of the different models (Section 4.1) and for the computation of the elasticity to price (Section 4.3). It contains 1868 observations. We describe now the different columns of the file:

ID The unique ID number of the individual.

SECTION_ID The ID number of the employee/class attribute of the individual. 1 corresponds to civil engineering students in their second year of bachelor, 2 corresponds to computer science students in their second year of bachelor, 3 corresponds to computer science in their first year of master, 4 to mathematics students in their first year of bachelor, 5 to employees, 6 to physics students in their first year of bachelor and 7 to life science students in their first year of bachelor.

DAY_YEAR The day number of the observation, with January 1 corresponding to 1.

STUDENT Binary variable with value 1 if the observation corresponds to a student, and 0 if it corresponds to an employee.

SEMESTER Count of the semester. First year of bachelor corresponds to semesters 1 and 2 and second year to semesters 3 and 4. The first year of master corresponds to semesters 7 and 8. Employees take value 0.

DAY_WEEK The day of the week, with 1 corresponding to Monday and 5 to Friday.

H_START The start hour of the observation, in 24-hour clock format.

M_START The start minute of the observation.

H_END The end hour of the observation, in 24-hour clock format.

M_END The end minute of the observation, in 24-hour clock format.

DURATION The duration of the observation, in minutes.

CHOICE The catering location choice, with value 1 for Cafe Le Klee, 2 for Cafeteria BC, 3 for Cafeteria BM, 4 for Cafeteria ELA, 5 for Cafeteria INM, 6 for Cafeteria MX, 7 for Cafeteria PH, 8 for L'Arcadie, 9 for L'Atlantide, 10 for Le Copernic, 11 for Le Corbusier, 12 for Le Giacometti, 13 for Le Parmentier, 14 for Le Vinci, 15 for L'Esplanade, 16 for L'Ornithorynque, 17 for Roulotte Diagonale, 18 for Roulotte Esplanade, 19 for Satellite, 20 for Self-service Le Hodler et 21 for Table de Vallotton.

MIN_PRICE_x The minimum price for a meal in the catering location x, in Swiss francs (CHF).

CAPACITY_INSIDE_x The indoor capacity of the catering location x, in number of seats.

CAPACITY_OUTSIDE_x The outdoor capacity of the catering location x, in number of seats.

OPEN_AV_x Binary variable with value 1 if the catering location x is open at the time of the observation, and 0 otherwise.

FOURCHETTE_VERTE_AV_x Binary variable with value 1 if the catering location x provides a special menu called "Fourchette verte", and 0 otherwise.

HOT_MEAL_AV_x Binary variable with value 1 if the catering location x sells hot meals for lunch, and 0 otherwise.

TERRACE_AV_x Binary variable with value 1 if the catering location x offers a terrace, and 0 otherwise.

SANDWICH_AV_x Binary variable with value 1 if the catering location x sells sandwiches, and 0 otherwise.

SERVICE_TABLE_AV_x Binary variable with value 1 if the catering location x offers table service, and 0 otherwise.

TAP_BEER_AV_x Binary variable with value 1 if the catering location x sells tap beer, and 0 otherwise.

DINNER_HOT_MEAL_AV_x Binary variable with value 1 if the catering location x sells hot meals for dinner, i.e., between 18:00 and 19:59, and 0 otherwise.

VISIBILITY_AV_x Binary variable with value 1 if the catering location x is visible from the main corridors and paths, and 0 otherwise.

EVALUATION_2013_x The evaluation of the catering location x, from a 2013 quality survey (takes value -1 if not available).

WORKSPACE_AV_x Binary variable with value 1 if the catering location x lets students work in it, and 0 otherwise.

CAFE_AV_x Binary variable with value 1 if the catering location x sells coffee, and 0 otherwise.

CAFE_PRICE_x The cost of a coffee in the catering location x, in Swiss francs (CHF).

RESTAURANT Binary variable with value 1 if the catering location x is classified as a restaurant, and 0 otherwise.

SELF Binary variable with value 1 if the catering location x is classified as a self service, and 0 otherwise.

CAFETERIA Binary variable with value 1 if the catering location x is classified as a cafeteria, and 0 otherwise.

CARAVAN Binary variable with value 1 if the catering location x is classified as a food truck, and 0 otherwise.

OTHER Binary variable with value 1 if the catering location x is not classified as any of the previous 4 types, and 0 otherwise.

FIDELITY_CARD_x Binary variables with value 1 if the catering location x offers a fidelity card, and 0 otherwise.

SELECTA_AV_x Binary variable with value 1 if the catering location x includes a vending machine, and 0 otherwise.

MICROWAVE_AV_x Binary variable with value 1 if the catering location x offers a microwave to heat your own food, and 0 otherwise.

DISTANCE_x The distance from the previous activity episode, using the pedestrian graph described in Danalet et al. (2014) and available in Danalet, Antonin (2015), in meters (takes value -1 if not available).

TEMPERATURE The average temperature of the day of the observation, in degree Celsius.

MAX_TEMP The maximum temperature of the day of the observation, in degree Celsius.

RAIN The rainfall for the day of the observation, in millimeter.

SUNNY_DAY_AV A binary variable with value 1 if no rain was recorded, 0 otherwise.

SUN_AND_HEAT_MIN_20 Binary variable with value 1 if no rain was recorded and the maximum temperature is higher than 20 °C in the day of the observation, 0 otherwise.

MOST_CHOSEN_x Binary variable with value 1 if the most frequently visited catering location for lunch, i.e., between 11:30 and 13:59, is similar to the current observation, and 0 otherwise (takes value -1 if not available). The frequency of visits for the given individual is computed based only on past visits (as compared to the current observation) and not on the full history. In the case of multiple catering locations with the same maximum number of visits in the past, the most frequently visited location is randomly selected among them.

HOURS_FROM_PREVIOUS_CHOICE The number of hours between the current observation at lunch time, i.e. between 11:30 and 13:59, and the previous catering location choice at lunch time (takes value -1 if not available).

PREVIOUS_CHOICE_MORNING_TRUE_x Binary variable with value 1 if the previous catering location choice made in the morning, i.e., between 7:00 and 11:29, is similar to the current observation, and 0 otherwise (takes value -1 if not available).

PREVIOUS_CHOICE_AFTERNOON_TRUE_x Binary variable with value 1 if the previous catering location choice made in the afternoon, i.e., between 14:00 and 22:00, is similar to the current observation, and 0 otherwise (takes value -1 if not available).

FIRST_CHOICE_MORNING_TRUE_x Binary variable with value 1 if the first catering location choice (first observation) made by the individual in the morning, i.e., between 7:00 and 11:29, is similar to the current observation, and 0 otherwise (takes value -1 if not available).

FIRST_CHOICE_AFTERNOON_TRUE_x Binary variable with value 1 if the first catering location choice (first observation) made by the individual in the afternoon, i.e., between 14:00 and 22:00, is similar to the current observation, and 0 otherwise (takes value -1 if not available).

MOST_CHOSEN_MORNING_x Binary variable with value 1 if the most frequently visited catering location in the morning, i.e., between 7:00 and 11:29, is similar to the current observation, and 0 otherwise (takes value -1 if not available). The frequency of visits for the given individual is computed based only on past visits (as compared to the current observation) and not on the full history. In the case of multiple catering locations with the same maximum number of visits in the past, the most frequently visited location is randomly selected among them.

2.2 Datasets for validation

`/data/2Validation/`

These two datasets are used for the validation of the different models (Section 4.2). They contain only the data corresponding to the morning and the lunch break.

`/data/2Validation/validation_dataset_past_obs_lunch.dat`

The estimation dataset, `validation_dataset_past_obs_lunch.dat`, is used for the estimation of the different models (Section 4.2, in folders `1Estimation`) using only the past observations for each individuals. It contains 1512 observations. The different columns of the file are similar to the ones in Section 2.1.

`/data/2Validation/validation_dataset_most_recent_obs_lunch.dat`

The validation dataset, `validation_dataset_most_recent_obs_lunch.dat`, is used to apply the different models with the parameter estimates from the previous step (Section 4.2, in folders `2Simulation`). This dataset contains only the most recent observation for each individual (i.e., 144 observations for the morning and the lunch break). The different columns of the file are similar to the ones in Section 2.1.

2.3 Dataset for forecasting

`/data/4Forecasting/val_forecast_all_day_20_epicure.dat`

This dataset contains the full set of data (Section 2.1), plus information about the new alternative, the catering location *L'Epicure*, coded as $x = 22$. It contains 175 observations, corresponding to 175 different individuals, for the full day.

3 Models

`/models/`

The model specification files have an extension `.py`. Their syntax is based on the Python programming language, with extensions for the specific needs of Pythonbiogeme. For more information about this syntax and how to use Pythonbiogeme, check `http://biogeme.epfl.ch`.

3.1 Estimation

`/models/1Estimation/`

First, the parameters to be estimated are defined. The alternative specific constants are named `ASC` with the abbreviation of the name of the catering locations.

Then, we define new variables in addition to the variables defined in the data file.

The utility functions are defined similarly to what is described in Danalet et al. (2016).

Note in the case of models with agent effect the creation of normally distributed random variable using the `bioNormalDraws` command. A different set of draws will be generated for each group in the data file. Here, the groups are identified by `ID`, and a set of draws is generated for each individual (and not for each observation).

The method proposed by Wooldridge (2005) is implemented by creating a parameter `ALPHA_FIRST_LUNCH_CHOICE` and by adding in the utility function the term:

```
ALPHA_FIRST_LUNCH_CHOICE * first_choice_filter_x
```

for each catering location alternative x, where `first_choice_filter_x` has value 1 if catering location choice was chosen at the first observation, and 0 otherwise.

3.2 Validation

`/models/2Validation/`

For the validation, the parameters are fixed to their estimated value using the model for estimation (Section 3.1) with the dataset containing only the past observations (Section 2.1). These results are described in Section 4.2 and have been copied in the beginning of model specification file.

3.3 Elasticity

`/models/3Elasticity/`

For the validation, the parameters are fixed to their estimated value using the model for estimation (Section 3.1) with the full dataset (Section 2.2). These results are described in Section 4.1 and have been copied in the beginning of model specification file.

The elasticity is computed for students and employees for each catering location. The formula used for the simulation is:

```
logitelas1 = OPEN_AV_1 * (1.0 - prob1) * lunch_price_min_1 * BETA_PRICE
```

where `OPEN_AV_1` is a binary variable with value 1 if the catering location is open and 0 otherwise, `prob1` is the choice probability for alternative 1, `lunch_price_min_1` is the cost variable (i.e., the cost of a meal for lunch in catering location 1), and `BETA_PRICE` is the cost parameter. It corresponds to the elasticity $E_{x_{ink}}^{P_n(i)}$ of the probability of an individual n choosing alternative i with respect to a change in attribute k:

$$\begin{aligned} E_{x_{ink}}^{P_n(i)} &= \frac{\partial P_n(i)}{\partial x_{ink}} \frac{x_{ink}}{P_n(i)} \\ &= \beta_k P_n(i)\big(1 - P_n(i)\big)\frac{x_{ink}}{P_n(i)} \\ &= \beta_k\big(1 - P_n(i)\big)x_{ink} \end{aligned}$$

In the case of models with agent effect, an extra step is needed to compute a simulated loglikelihood function. In the code, `P1` is used instead of `prob1`, where `P1` is defined as:

```
P1 = mixedloglikelihood(prob1)
```

3.4 Forecasting

`/models/4Forecasting/`

For the validation, the parameters are fixed to their estimated value using the model for estimation (Section 3.1) with the full dataset (Section 2.2). These results are described in Section 4.1 and have been copied in the beginning of model specification file.

Note that there is a 22^{nd} utility function for the new catering location. Its abbreviation is `HC`. It is similar to utility function `V12` corresponding to *Le Giacometti*, except that it uses the variables corresponding to alternative 22 corresponding to *L'Epicure*.

The nest structure described in Danalet et al. (2016) is described in the model specification files by creating two nests, `nongia` and `gia`. The `gia` nest contains alternatives 12 and 22, corresponding to *L'Epicure* and *Le Giacometti*, i.e., the new alternative and its most similar existing alternative. The `gia` nest is associated to a nest parameter `MU`. The second nest, `nongia`, contains all other alternatives. The `nongia` nest is associated to a nest parameter of 1.0.

For each model, 4 different model specification files are available, for $\theta = 1, 2, 5, 10$ (see Danalet et al.; 2016).

4 Results

4.1 Estimation

`/results/1Estimation/`

The results of the estimation for the different models are generated using the full dataset (Section 2.1) and the estimation model specification files (Section 3.1). The result folders contain results as HTML files, LaTeX files and parameter files in python format, particularly useful to copy the outcomes of estimation in the model specification files for elasticities (Section 3.3) and for forecasting (Section 3.4)

4.2 Validation

`/results/2Validation/`

For each model, the results of the validation are decomposed in results of the estimation of the different models (folder `/1Estimation/`) and in results of the simulation of the different models (folder `/2Simulation/`).

The estimation is performed using the dataset containing only past observations (Section 2.2) with the estimation model specification files for estimation (Section 3.1).

The simulation is performed using the dataset containing the most recent observations (Section 2.2) with the simulation model specification files for validation (Section 3.2). The folder `/2Simulation/` contains the raw output of Pythonbiogeme in HTML format and a spreadsheet for further manipulations of the output. The `.udraws` files containing the draws used in the cases with agent effect are not provided here, in order to keep the size of the folder reasonable. They can be provided on request, for exact reproduction of the results. Anyway, we used the default seed to generate these results. The manipulations performed with OpenOffice consists in computing the sum of the squares of the errors, comparing the predicted number of visitors with the observed number of visitors in each catering location.

Note that in the cases with an agent effect, the output of Pythonbiogeme is the simulated loglikelihood and not the choice probabilities. In order to get the choice probabilities, we compute the exponential of the simulated loglikelihood in a spreadsheet.

4.3 Elasticity

`/results/3Elasticity/`

The simulation is performed using the full dataset (Section 2.1) with the simulation model specification files for the computation of the elasticities (Section 3.3).

The main output is the aggregate `Average (non zeros)` for each catering location. It averages all values different from zero for each catering location (i.e., column) over the observations (i.e., rows). Values of zero means either that the individual is a student when computing the elasticities for employees (or conversely, the individual is an employee when computing the elasticities for students), or that the catering location was closed when the individual made the decision, or that this catering location does not provide food and therefore has a cost value of zero. Therefore, values of zero are excluded when computing the aggregate average elasticities.

The `.udraws` files containing the draws used in the cases with agent effect are not provided here, in order to keep the size of the folder reasonable. They can be provided on request, for exact reproduction of the results. Anyway, we used the default seed to generate these results.

4.4 Forecasting

`/results/4Forecasting/`

The simulation is performed using the full dataset with information about the new alternative (Section 2.3) with the simulation model specification files for forecasting (Section 3.4).

The main output is the aggregate `Average` for the new catering location, `HC`. It provides the predicted average frequency of visits.

The `.udraws` files containing the draws used in the cases with agent effect are not provided here, in order to keep the size of the folder reasonable. They can be provided on request, for exact reproduction of the results. Anyway, we used the default seed to generate these results.

Note that in the cases with an agent effect, the output of Pythonbiogeme is the simulated loglikelihood and not the choice probabilities. In order to get the choice probabilities, we compute the exponential of the simulated loglikelihood in a spreadsheet.

References

Bierlaire, M. (2003). BIOGEME: a free package for the estimation of discrete choice models, *Proceedings of the 3rd Swiss Transportation Research Conference*, Monte Verità, Ascona, Switzerland.
URL: *http://www.strc.ch/2003/bierlaire.pdf*

Bierlaire, M. and Fetiarison, M. (2009). Estimation of discrete choice models: extending BIOGEME, *Swiss Transport Research Conference (STRC)*, Monte Verità, Ascona, Switzerland.
URL: *http://www.strc.ch/2009/Bierlaire_ 3.pdf*

Danalet, A. (2015). Activity choice modeling for pedestrian facilities.
URL: *https://doi.org/10.5075/epfl-thesis-6806*

Danalet, A., Farooq, B. and Bierlaire, M. (2014). A Bayesian approach to detect pedestrian destination-sequences from WiFi signatures, *Transportation Research Part C* **44**: 146–170.
URL: *http://dx.doi.org/10.1016/j.trc.2014.03.015*

Danalet, A., Tinguely, L., Lapparent, M. d. and Bierlaire, M. (2016). Location choice with longitudinal WiFi data, *Journal of Choice Modelling* **18**: 1–17.
URL: *https://dx.doi.org/10.1016/j.jocm.2016.04.003*

Danalet, Antonin (2015). A Bayesian Approach to Detect Pedestrian Destination-Sequences from WiFi Signatures: Data (Transp. Res. Part C, 2014).
URL: *http://dx.doi.org/10.5281/zenodo.15798*

Wooldridge, J. M. (2005). Simple solutions to the initial conditions problem in dynamic, nonlinear panel data models with unobserved heterogeneity, *Journal of Applied Econometrics* **20**(1): 39–54.
URL: *http://dx.doi.org/10.1002/jae.770*