D200, Problem Set 2: Discrete Choice Models

Due: 19 February 2026 here in groups of up to 2.

Stefan Bucher

This problem set will review classification as discussed in the lecture through the lens of discrete choice modeling, a classically used method in economics.

The problem set uses the choice-learn package, see here for more background.

Problem 1: The Conditional Logit Model

Discrete choice models are built on the **Random Utility Maximization (RUM)** framework. A decision-maker chooses the alternative with the highest utility from a set of available options. The utility of alternative j for individual i is:

$$U_{ij} = V_{ij} + \varepsilon_{ij}$$

where V_{ij} is the **systematic (observable) utility** and ε_{ij} is a **random error term** capturing unobserved factors.

The **Conditional Logit** model assumes:

1. Utility is linear in attributes: $V_{ij} = \sum_k \beta_{ik} \cdot x_{jk}$
2. Errors are i.i.d. Type I Extreme Value (Gumbel) distributed

The probability of individual i choosing alternative j from choice set $\mathcal{A}$ is then given by

$$P_{ij} = \frac{\exp\left(\sum_k \beta_{ik} \cdot x_{jk}\right)}{\sum_{a \in \mathcal{A}} \exp\left(\sum_k \beta_{ik} \cdot x_{ak}\right)}$$

The ModeCanada Dataset

We'll work with the **ModeCanada** dataset, which contains transportation choices for intercity trips between Montréal and Toronto. This is a classic dataset in choice modeling research.

(1a) Load the ModeCanada dataset and explore its structure:

```
from choice_learn.datasets import load_modecanada
transport_df = load_modecanada(as_frame=True)
print(f"Dataset shape: {transport_df.shape}")
display(transport_df.head(8))
```

```
Dataset shape: (15520, 11)
```

	case	alt	choice	dist	cost	ivt	ovt	freq	income	urban	noalt
0	1	train	0	83	28.25	50	66	4	45.0	0	2
1	1	car	1	83	15.77	61	0	0	45.0	0	2
2	2	train	0	83	28.25	50	66	4	25.0	0	2
3	2	car	1	83	15.77	61	0	0	25.0	0	2
4	3	train	0	83	28.25	50	66	4	70.0	0	2
5	3	car	1	83	15.77	61	0	0	70.0	0	2
6	4	train	0	83	28.25	50	66	4	70.0	0	2
7	4	car	1	83	15.77	61	0	0	70.0	0	2

The data is in **long format**: each row represents one alternative within a choice situation. Key columns:

- `case`: identifies each choice situation (one traveler's decision)
- `alt`: the transportation mode (train, air, bus, car)
- `choice`: 1 if this alternative was chosen, 0 otherwise
- `cost`, `ivt` (in-vehicle time), `ovt` (out-of-vehicle time), `freq` (frequency): alternative attributes
- `income`: traveler characteristic (same across alternatives within a case)

Examine a single choice situation by filtering for `case == 1`. How many alternatives were available? Which was chosen?

(1b) The `ChoiceDataset` is choice-learn's core data structure. It organizes:

- **Choices**: which alternative was selected
- **Items features**: attributes that vary by alternative (cost, time, etc.)
- **Shared features**: attributes that are constant across alternatives (income, etc.)

Convert the DataFrame to a `ChoiceDataset`:

```
from choice_learn.data import ChoiceDataset

canada_dataset = ChoiceDataset.from_single_long_df(
    df=transport_df,
    items_id_column="alt",           # identifies each alternative
    choices_id_column="case",         # identifies each choice situation
    choices_column="choice",          # indicates which was chosen
    shared_features_columns=["income"],  # traveler characteristics
    items_features_columns=["cost", "freq", "ovt", "ivt"],  # alternative attributes
    choice_format="one_zero"
)

print(canada_dataset.summary())
```

```
%=====================================================================%
%%% Summary of the dataset:
%=====================================================================%
Number of items: 4
Number of choices: 4324
%=====================================================================%
 Shared Features by Choice:
 1 shared features
 with names: (['income'],)


 Items Features by Choice:
4 items features
 with names: (['cost', 'freq', 'ovt', 'ivt'],)
%=====================================================================%
```

Model Specification

(1c) The key modeling decision is specifying the utility function. For ModeCanada, consider:

$$U_{ij} = \beta_j^{inter} + \beta^{cost} \cdot \text{cost}_j + \beta^{freq} \cdot \text{freq}_j + \beta^{ovt} \cdot \text{ovt}_j + \beta_j^{ivt} \cdot \text{ivt}_j + \beta_j^{income} \cdot \text{income}_i$$

Note the subscripts:

- β^{cost}, β^{freq}, β^{ovt} are **shared** coefficients (same effect for all modes)

- β_j^{ivt}, β_j^{income}, β_j^{inter} are **alternative-specific** (different for each mode)

Why might we want different coefficients for in-vehicle time across modes? (Think about the experience of traveling by train vs. car vs. plane.)

(1d) Implement and fit the Conditional Logit model from (1c) using choice-learn's `ConditionalLogit` class. Use the utility specification above, with `optimizer="lbfgs"` and `get_report=True`.

Hints:

- Use `add_shared_coefficient()` for coefficients that are the same across all alternatives, and `add_coefficients()` for alternative-specific ones.
- For alternative-specific constants (intercept, income), you must normalize one alternative to zero. Why?

(1e) Interpret the estimated coefficients:

1. What is the sign of β^{cost}? Does this make economic sense?
2. Compare the intercepts across modes. Which mode has the highest "baseline" utility?
3. How do the income coefficients vary? What does this tell us about mode choice and income?

(1f) Price Elasticity measures how choice probabilities change with price. For the logit model:

$$\eta_{jj} = \frac{\partial P_{ij}}{\partial p_j} \cdot \frac{p_j}{P_{ij}} = \beta^{cost} \cdot p_j \cdot (1 - P_{ij})$$

This is the **own-price elasticity**. Compute it for the car alternative at the mean values.

Problem 2: RUMnet — Neural Network Choice Models

The Conditional Logit assumes utility is *linear* in attributes. **RUMnet** (Aouad & Désir, 2022) relaxes this assumption using neural networks while maintaining the RUM framework.

(2a) For this problem, we'll use the more complex **Expedia** hotel booking dataset. First download `train.csv` from Kaggle and save it to your Python environment's `choice_learn/datasets/data/expedia.csv` (if the path is wrong, `choice_learn` will tell you the exact location in a `FileNotFoundError`).

Load the dataset using `load_expedia(as_frame=False, preprocessing="rumnet")`, keep only the first 5000 choices for speed, and split 80/20 into training and test sets. Explore the

dataset structure — how many choices, items, and features does it have? What do the choice set sizes look like?

(2b) Write down a sensible model specification for the Conditional Logit model for the Expedia dataset, for example using the hotel features: log(price), star rating, review, whether the hotel is a brand, location desirability scores. You may also want to include hotel fixed effects. Fit your model and report the cross-entropy loss on the test data using TensorFlow's `tf.keras.losses.CategoricalCrossentropy`.

(2c) Display the resulting parameter estimates and interpret them. What is the sign of the price coefficient? Which features matter most?

(2d) Now fit the **RUMnet** model shipped with `choice_learn` to the Expedia dataset. The dataset has 46 product features and 84 customer features. Report the cross-entropy loss on the test data and compare it to the Conditional Logit.

(2e) Discuss: What are the tradeoffs between Conditional Logit and RUMnet?