Decision Making Project Report

Clovis Piedallu Basile Heurtault Jérémy Mathet

February 6, 2025

Contents

Introduction

In this project, we explore preference learning, a key challenge in modern decision analysis and recommendation systems. Our focus is on learning utility functions from pairwise preferences using Mixed Integer Programming (MIP). The goal is to develop models that can effectively capture and predict decision-making patterns from observed preference data, particularly in scenarios where different groups of decision-makers may exhibit distinct preference behaviors.

Building upon the foundational UTA (UTilités Additives) method introduced by Jacquet-Lagrèze and Siskos[1],

[1] Jacquet-Lagrèze, E., & Siskos, J. (1982). Assessing a set of additive utility functions for multicriteria decision-making, the UTA method. European Journal of Operational Research, 10(2), 151-164.

our work extends the original approach to handle multiple preference clusters simultaneously. The UTA method provides a framework for inferring additive utility functions from preference rankings, and has become a cornerstone in multicriteria decision analysis.

The project is structured in three main parts:

1. Definition and formulation of a MIP model to learn piecewise-linear utility functions with multiple clusters
2. Implementation and analysis of the MIP model
3. Development of a heuristic approach for preference learning to handle larger-scale problems

Our approach builds upon the UTA (UTilités Additives) method while extending it to handle multiple preference clusters simultaneously.

1. Problem formulation

Objective function

We seek to minimize :

$$\min \sum_{k=1}^{K} \sum_{j=1}^{P} \sigma^k(j)$$

Decision variables

- $u_i^k(x_i^l) \geq 0$: Continuous decision variable $\quad \forall i \in [1, n], \quad \forall l \in [1, L], \quad \forall k \in [1, K]$
- $c_k^{(j)} \in \{0, 1\}$: Binary decision variable $\quad \forall j \in [1, P], \quad \forall k \in [1, K]$

Constraints

The problem is subject to the following constraints:

Normalization constraints

$$u_i^k(x_i^0) = 0 \quad \forall i \in [1, n], \forall k \in [1, K]$$

$$\sum_{i=1}^{n} u_i^k(x_i^L) = 1 \quad \forall k \in [1, K]$$

Monotonicity constraints

$$u_i^k(x_i^{l+1}) - u_i^k(x_i^l) \geq \epsilon \quad \forall l \in [0, L]$$

Preference constraints

$$M(c_k^{(j)} - 1) \leq \sum_{i=1}^{n} [u_i^k(x_i^{(j)}) - u_i^k(y_i^{(j)})] + \sigma^k(x^{(j)}) \leq M c_k^{(j)} \quad \forall k \in [1, K], \forall j \in [1, P]$$

$$\sum_{k=1}^{K} c_k^{(j)} \geq 1 \quad \forall j$$

Domain constraints

$$0 \leq u_i^k(x_i^l) \leq 1 \quad \forall i, \forall k, \forall l$$

$$c_k^{(j)} \in \{0, 1\} \quad \forall j, \forall k$$

2. Implementation and results

2.1 Model implementation

It's an extension of UTA method considering clustering in addition.

The MIP model was implemented using the Pyomo optimization framework with CPLEX as the solver. It's an extension of UTA method considering clustering in addition.

2.2 Results and visualization

Below are the visualization results showing the learned utility functions for each feature across different clusters:

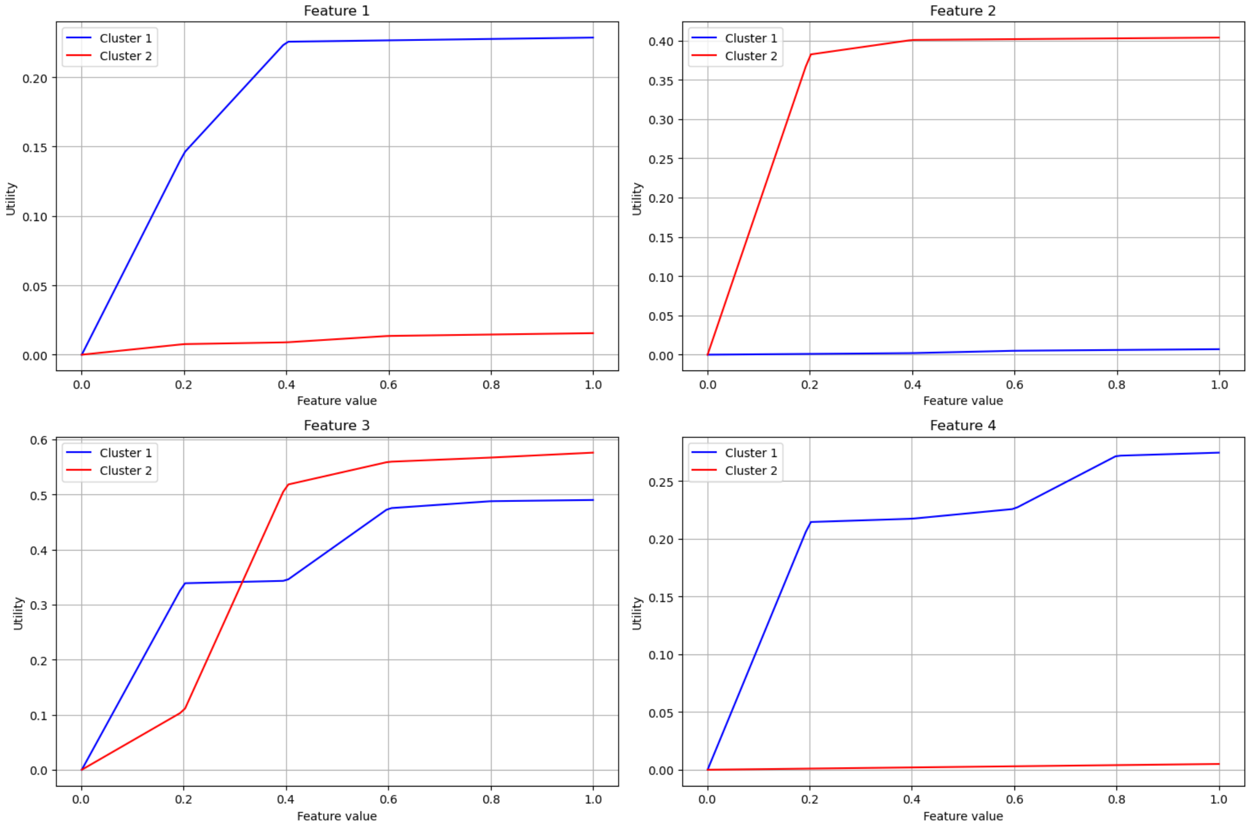

Figure 1: *Learned piecewise-linear utility functions for each feature and cluster*

3. Heuristic approach for cluster-based decision functions

3.1 Overview of the Heuristic approach

The goal of this heuristic approach is to estimate decision functions that allocate preference pairs ((X, Y)) into (K) clusters and iteratively refine the assignments based on utility differences. Instead of relying on an exact Mixed Integer Programming (MIP) formulation, we employ an adaptive heuristic that:

1. **Randomly assigns initial clusters** to each preference pair.
2. **Computes utility functions** based on piecewise linear functions.
3. **Refines clusters iteratively** by minimizing the objective function.

4. **Stops when cluster assignments stabilize**, indicating convergence.

This method is an approximation and does not guarantee an optimal solution, but it efficiently finds a structured decision function that respects user preferences.

3.2 Modifications to the car dataset

The dataset initially contained a mix of categorical and numerical data. To prepare it for processing, several transformations were applied:

3.2.1 Extracting preference pairs

Each row in the dataset represents a customer who selected a vehicle from six possible options. - The column `choice` indicates the selected vehicle. - The other five alternatives are considered **rejected choices**. - From this, we generated preference pairs ((X, Y)), where (X) is the chosen vehicle and (Y) is a rejected one.

3.2.2 Handling categorical data

The dataset contained categorical features such as `type` (e.g., sedan, SUV, van) and `fuel` (e.g., gasoline, electric, hybrid). These were converted into numerical form using **one-hot encoding**, ensuring that the dataset was fully numerical.

3.2.3 Normalization and cleaning

- We ensured that all features were properly scaled and cleaned.
- All numerical transformations were validated to avoid errors during computation.
- Any remaining categorical values were converted to numeric types.

These steps ensured that the dataset could be directly used in the heuristic model.

Extracted 23270 preference pairs.

```
# Sample of X data (first few rows and last few rows):
[
    ['van', 'regcar', 'van', ..., 4.8177056, 5.1388859, 5.1388859],
    ['van', 'regcar', 'van', ..., 4.8177056, 5.1388859, 5.1388859],
    ['van', 'regcar', 'van', ..., 4.8177056, 5.1388859, 5.1388859],
    ...
    ['regcar', 'van', 'regcar', ..., 5.1388859, 4.1753448, 4.1753448],
    ['regcar', 'van', 'regcar', ..., 5.1388859, 4.1753448, 4.1753448],
    ['regcar', 'van', 'regcar', ..., 5.1388859, 4.1753448, 4.1753448]
]

# Sample of Y data (first few rows and last few rows):
[
    ['regcar', 'cng', 4.1753448],
    ['van', 'electric', 4.8177056],
    ['stwagon', 'electric', 4.8177056],
    ...
    ['stwagon', 'cng', 5.1388859],
    ['regcar', 'gasoline', 4.1753448],
    ['truck', 'gasoline', 4.1753448]
]
```

Finally we dropped the categorical values because we encountered issues with the One-hot-encoding. We only kept numerical ones (cf loading_cars_data).

This function ensures that the dataset is structured appropriately for preference-based decision modeling.

3.3 Implementation of the Heuristic model

3.3.1 Initialization

Unlike MIP-based optimization, we initialize our clusters randomly:
- Each preference pair (X_j, Y_j) is assigned to one of K clusters randomly
- Initial cluster assignments $c_j^k \in \{0,1\}$ form the starting point

3.3.2 Utility function definition

For each cluster $k \in \{1, ..., K\}$:
- Utility function $u^k : \mathbb{R}^n \to \mathbb{R}$ is piecewise linear
- $u^k(x) = \sum_{i=1}^n u_i^k(x_i)$ where each u_i^k is a piecewise linear function
- Functions initialized randomly and refined through iterations

3.3.3 Iterative refinement

Repeat until convergence:

1. For each pair (X_j, Y_j), compute utility differences:

$\Delta_j^k = u^k(X_j) - u^k(Y_j)$ for all k

2. Update cluster assignments:

$$c_j^k = \begin{cases} 1 & \text{if } k = \arg\max_k \Delta_j^k \\ 0 & \text{otherwise} \end{cases}$$

3. Stop when $\sum_{j,k} |c_j^k(t) - c_j^k(t-1)| = 0$

4. Results and observations

4.1 Convergence Properties

- Model converges in $O(T)$ iterations where $T \ll$ problem size
- Objective function $\sum_{j,k} c_j^k \Delta_j^k$ monotonically improves
- Final clusters exhibit stable preference patterns

4.2 Performance analysis

Strengths

- $O(n)$ complexity vs $O(2^n)$ for exact MIP
- Scales linearly with dataset size
- Independent of commercial solvers

Limitations

- Local optima due to random initialization
- No global optimality guarantees
- Sensitive to hyperparameters:
  - Number of clusters K
  - Number of pieces L in piecewise functions
  - Maximum iterations T