UTA-NM:
Explaining Stated Preferences with Additive Minimum Non-Monotonic Utility Functions

Tomáš Kliegr

Department of Information and Knowledge Engineering, University of Economics, Prague, Czech Republic

Outline

- Introduction to the UTA method
  - Not well known in Preference Learning community (according to one reviewer)
- Motivation for the non-monotonic extension
- Illustrative Example
- Limitations and further work

VŠE PRAHA

Example

- g_1 *looks*, g_2 *wittiness*, g_3 *sport attitude*
- Alternatives $a_1 \ldots a_5$
- The DM preferences:

Stated Preference	Name	Look	Wittiness	Sport
1.	John	4	2	1
2.	Ashley	2	1	2
3.	Peter	3	3	3
4.	Martin	2	4	4
5.	Stan	2	4	5

Legend: 1(Low) … 5 (High)

VŠE PRAHA

UTA Method

- UTA Method is a **linear-programming method** for disaggregation-aggregation analysis of preferences.
- Input for the method are implicit preferences in the form of the order of alternatives. E.g. $a_1 > a_2 \sim a_3 > a_4 > a_5$
- Alternatives are described by a set of criteria $g_1,\dots,g_n$
- The utility from an alternative is given by the sum of utilities from the criteria:

$$u(a) = \sum_{i=1}^{n} u_i(g_i(a))$$

Name	Look	Wittiness	Sport
John	4	2	1

$$u(John) = u_{looks}(4) + u_{wittines}(2) + u_{sport}(1)$$

VŠE PRAHA

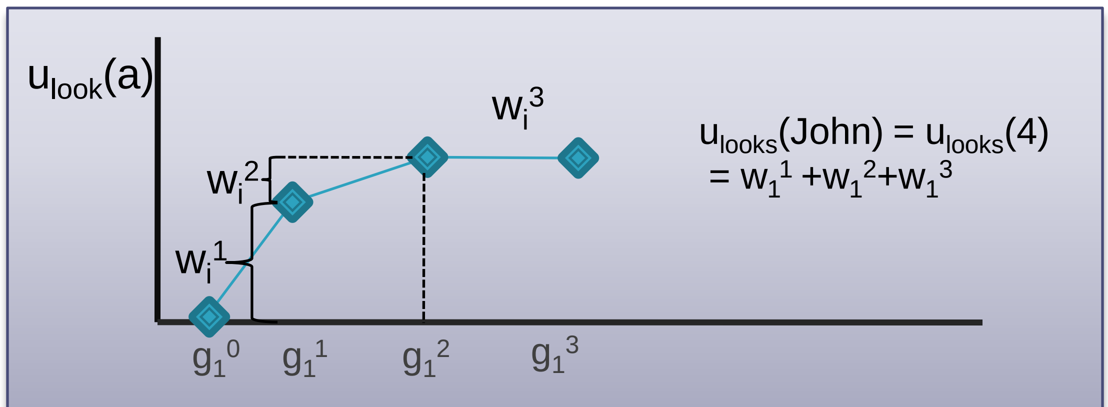
Partial Utility Function

- The value at breakpoints of partial utility function u_i is given by the sum of marginal utilities w_i^j

$$u_i(g_i^0) = 0$$

$$j = 1 \dots \varphi_i$$

$$u_i(g_i^j) = \sum_{t=1}^{j} w_{it}$$



- Partial utility functions in traditional UTA methods are monotonic
- UTA finds values of marginal utility variables w_i^j that generate the most similar ranking to the reference ranking

VŠE PRAHA

Infering marginal utilities from the ranking

- Introduce two errors σ^+ and σ^- for each alternative
- Susbtract utilities of consecutive alternatives:

$$\Delta(a_k, a_{k+1}) = u[\boldsymbol{g}(a_k)] - \sigma^+(a_k) + \sigma^-(a_k) - u[\boldsymbol{g}(a_{k+1})] + \sigma^+(a_{k+1}) - \sigma^-(a_{k+1})$$

$$k = 1,2,\ldots K-1 \begin{cases} \Delta(a_k, a_{k+1}) > \delta \text{ if } a_k > a_{k+1} \\ \Delta(a_k, a_{k+1}) = 0 \text{ if } a_k \sim a_{k+1} \\ \sigma^+(a_k), \sigma^-(a_k) \geq 0 \end{cases}$$

- Objective function minimizes the sum of errors σ^+ and σ^-

A1: u(John) = $w_1^1 + w_1^2 + w_1^3 + w_2^1$ A2: u(Ashley) = $w_1^1 + w_3^1$

Since Rank(John) = 1 and Rank(Ashley)=2 then

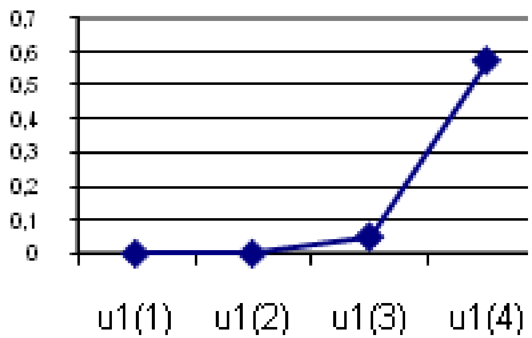
$$d(A1,A2) = w_1^2 + w_1^3 + w_2^1 - w_3^1 - \sigma_1^+ + \sigma_1^- + \sigma_2^+ - \sigma_2^- > \delta, \delta = 0.05$$

Errors σ^+ and σ^- allow UTA to find imperfect solutions

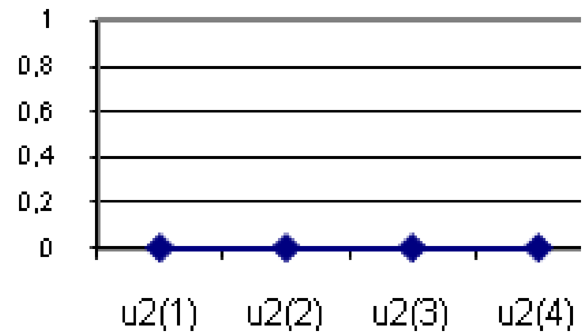
Example

Pref	Name	Look	Wittiness	Sport
1	John	4	2	1
2	Ashley	2	1	2
3	Peter	3	3	3
4	Martin	2	4	4
5	Stan	2	4	5

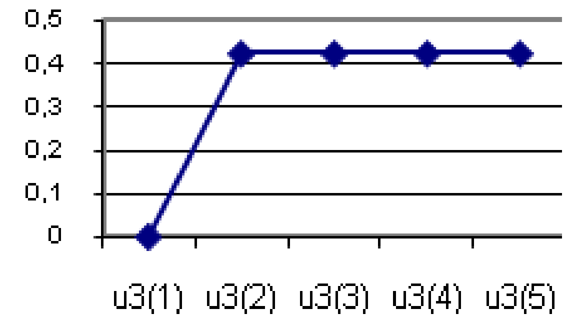
- Explanation for stated order
  - $a_1 > a_2 > a_3 > a_4 > a_5$



Looks



Wittiness



Attitude to sport

By applying the model back to data we get:

$$a_1 > a_2 = a_3 = a_4 = a_5$$

The discovered solution does not fully comply with the stated order of the alternatives

VŠE PRAHA

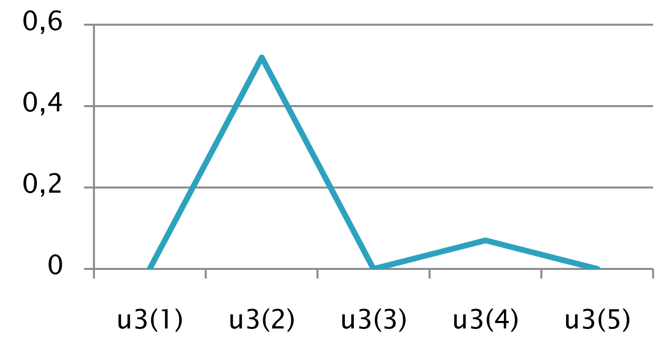
Motivation for non-monotonic UTA

- In the example, a fully fitting model was not found because the preferences of the DM were actually non-monotonic in criterion „g3: Sport“
- UTA assumes monotonic preferences
- The only non monotonic UTA algorithm (Despotis, Zopounidis 93) has following limitations
  - The exact utility function shape need to be known beforehand
  - There is maximum one change of shape per utility function
  - Proposed fully non-monotonic version: UTA-NM

VŠE PRAHA

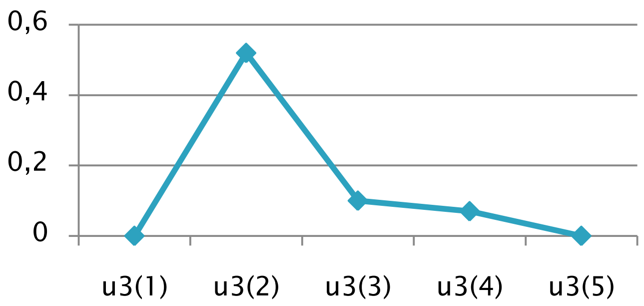
UTA – NM

- Inspired by UTA
- Relaxes the monotonicity assumption (by allowing negative marginal utility)
- Problem: solutions have many changes of shape in partial utility functions
  - overfitting
  - not interpretable



UTA – NM

- Inspired by UTA
- Relaxes the monotonicity assumption (by allowing negative marginal utility)
- Problem: solutions have many changes of shape in partial utility functions
  - overfitting
  - not interpretable



- UTA – NM **simultaneously** minimizes the **sum of errors** σ^+ and σ^- and the complexity of the model expressed by the **number of changes** in shape of partial utility functions.
- Challenge: **Keep the problem linear**

First approach: non-linear model, non-linear methods

- It is simple to count the number of changes in the shape if we can use nonlinear funcions such as *if* or *abs.*
- The work on linearized UTA-NM was preceded by experiments with non-linear methods Branch&Bound/GRG Solver
- Interval Global Solver – found optimal solution in (4h)
- Genetic algorithms
- These experiments were not successful. Best result Branch&Bound/GRG Solver initialized with UTA Star

VŠE PRAHA

LP program outline

$$[min]\ z = \sum_{a\in A_R} \sigma^{+}(a) - \sigma^{-}(a) + \mathrm{E}$$

s.t.

$$k = 1,2,\dots K-1 \begin{cases} \Delta(a_k, a_{k+1}) > \delta \ \text{if} \ a_k > a_{k+1} \\ \Delta(a_k, a_{k+1}) = 0 \ \text{if} \ a_k \sim a_{k+1} \\ \sigma^{+}(a_k), \sigma^{-}(a_k) \geq 0 \end{cases}$$

$$\forall i: min(u_i(g_i^0), u_i(g_i^1), \dots u_i(g_i^j), \dots, u_i(g_i^{\gamma_i})) = 0$$

$$\sum_{i=1}^{n} max(u_i(g_i^0), u_i(g_i^1), \dots u_i(g_i^j), \dots, u_i(g_i^{\gamma_i})) = 1$$

VŠE PRAHA

Utility calculation

$$g_i : A \rightarrow [g_{i^*}, g_i^*], g_i(a)$$

$$u(a) = \sum_{i=1}^{n} u_i\left(g_i\ (a)\right)$$

$$[g_i^0, g_i^1], \ldots, [g_i^{\gamma_{i-1}}, g_i^{\gamma_i}], g_i^* = g_i^{\gamma_i}$$

$$w_i^k = u_i\left(g_i^k\right) - u_i\left(g_i^{k-1}\right)$$

$$u_i\left(g_i^j\right) = \sum_{k=0}^{j} w_i^k \quad w_i^j, j = 0, \ldots \gamma_i$$

$$u'(a) = \sum_{i=1}^{n} u_i\left(g_i\ (a)\right) + \sigma^+(a) - \sigma^-(a)$$

VŠE PRAHA

Normalization

- UTA

$$u_i(g_i^0) = u_i(g_{i^*}) = w_i^0 = 0,$$

$$\sum_{i=1}^{n} u_i\left(g_i^{\gamma_i}\right) = \sum_{i=1}^{n} u_i\left(g_i^*\right) = 1$$

- UTA NM

$$min(u_i(g_i^0), u_i(g_i^1), \ldots u_i(g_i^j), \ldots, u_i(g_i^{\gamma_i})) = 0$$

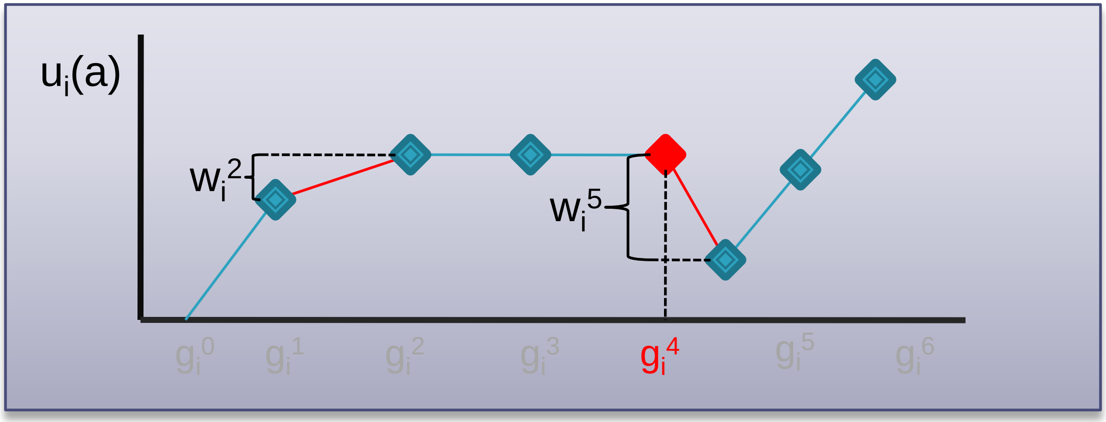
$$\sum_{i=1}^{n} max(u_i(g_i^0), u_i(g_i^1), \ldots u_i(g_i^j), \ldots, u_i(g_i^{\gamma_i})) = 1$$

More details in the paper

VŠE PRAHA

Penalization

- The value of the objective function is increased for each point in which the partial utility function u_i changes its shape
- The change of shape is detected from signs of marginal utilities w_i^l and w_i^{j+1} and is saved to binary variable e_i^j
  - w_i^l is nearest previous non-zero marginal utility



sign (w_i^l)	sign (w_i^l+1)	e_{ij}
−1	−1	0
−1	0	0
−1	+1	1
0	−1	0
0	0	0
0	+1	0
1	**−1**	**1**
+1	0	0
+1	+1	0

- Penalization element:

$$E = \sum_{i=1}^{n} \sum_{j=1}^{\gamma_i - 1} \mu_i^j \, \delta \, e_i^j$$

Locating the nearest previous nonzero w_i^l

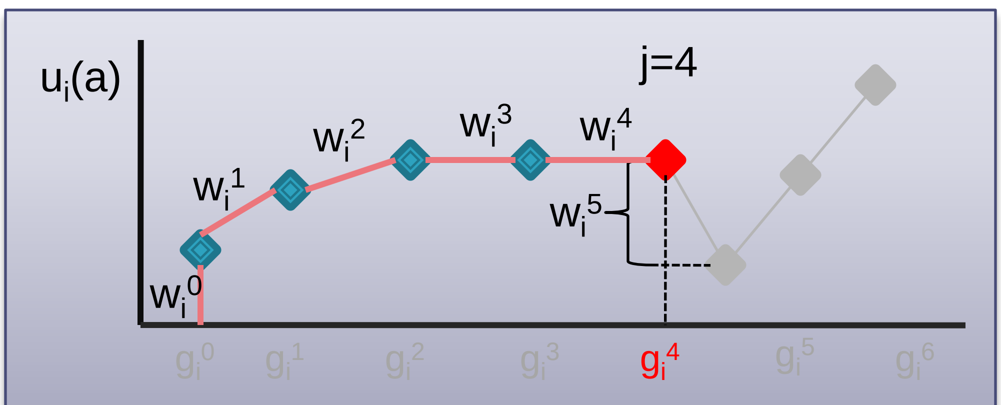
For g_i^4 (j=4) we search the nearest previous ***nonzero*** marginal utility w_i^l.

$$q = j, j-1, \dots, 0 \quad \begin{cases} \sum_{r=q}^{j} p_i^r + \sum_{r=q}^{j} n_i^r - \sum_{r=q}^{j} {}^r k_i^j \geq 0 \\ p_i^q + n_i^q - \sum_{r=q}^{j} {}^r k_i^j \leq 0 \end{cases}$$

$$\sum_{r=0}^{j} {}^r k_i^j \leq 1$$

sign(w_i)	p_i^r	y_i^r
+	1	0
–	0	1
0	0	0

Sign of marginal utility variable is expressed by the binary variables p_i^r, y_i^r .



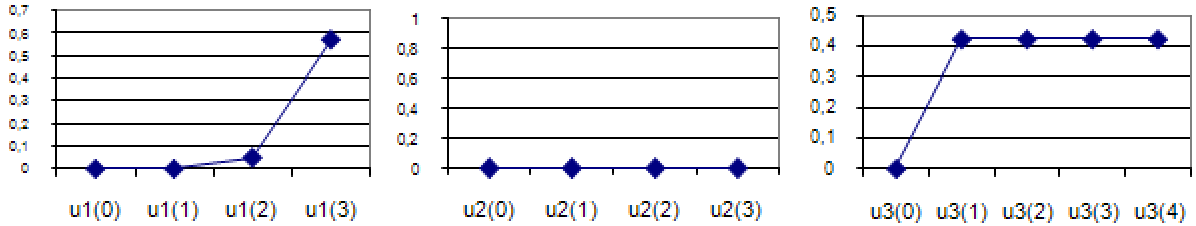
q	4	3	2	1	0
p_i^q	0	0	1	1	1
y_i^q	0	0	0	0	0
k_i^q	0	0	1	0	0
$\sum_{r=q}^{j=4} {}^r k_i^j$	0	0	1	1	1

Variable ${}^r k_i^q$ is set to 1 only iff w_i^q is the nearest previous nonzero utility to g_i^j

Example

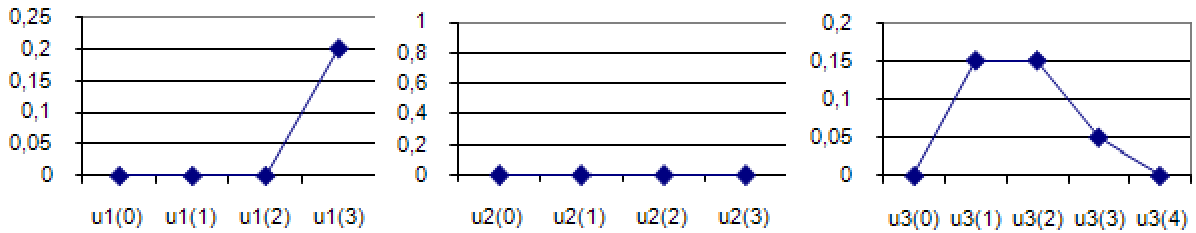
- Partner selection
- Explicit preferences :
  - the DM prefers middle value of sport endorsement
- Software used:
  - Frontline Premium Solver (commercial solver)
  - LP Solve (Open source solver)
  - Visual UTA 1.0 (Academic UTA Star implementation)

Information supplied



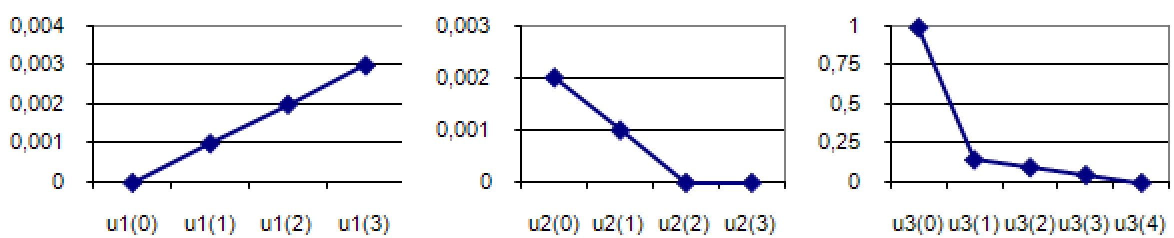
Utility functions found by **UTA Star**

Higher values are better
In all criteria



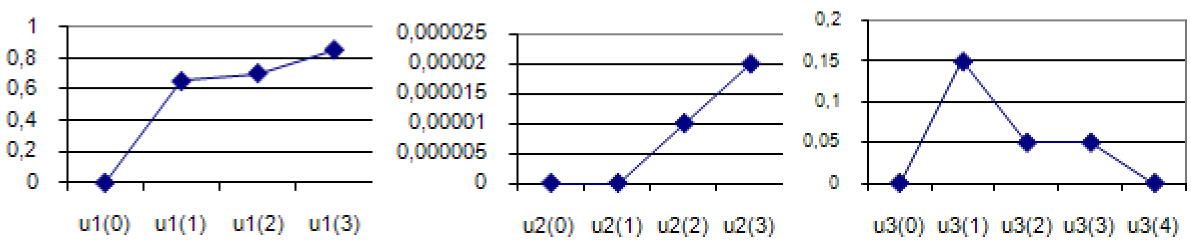
Utility functions found by **Despotis Method**

Higher values are better in u1 and u2, maximum of u3 is in u3(3)



Utility functions found by **UTA-NM**

None



Utility functions found by **UTA-NM with** $w_i^0 = 0$

The worst value (nonexclusive) is at the first breakpoint

Experimental results

- Model found with UTA-NM was the only one to fully match the stated order (Pearson coefficient equal to 1)
- The deviation from the explicitly expressed preferences is also small
- Performace-wise, UTA-NM was slowest with 40/20 seconds (LPSolve on T1 / RiskSolver on T2) compared to less than 1 sec for other methods.

Method	Final rank	Error sum	Pearson Coefficient	Local Extremes	Explicit preferences
DM	a1>a2>a3>a4>a5	NA	NA	NA	NA
UTA Star	a1>a2=a3=a4=a5	0,15	0,73	**0**	no
Despotis	a1>a2=a3=a4>a5	0,1	0,96	1	**yes/NA**
NM T1	**a1>a2>a3>a4>a5**	**0**	**1**	0	No
NM T2	**a1>a2>a3>a4>a5**	**0**	**1**	1	Partially

VŠE PRAHA

Conclusion

- UTA Star was generalized to work with non-monotonic preferences
- If there is a monotonic solution fully complaint with stated preferences exists, UTA-NM outputs it
- UTA-NM has means to prevent overfitting
- Expert can input prior knowledge about the shape of the utility functions, these are used to weight contribution of changes in shape to the objective f.
- Resulting problem is linear and convex and hence processable with standard LP solvers

- Further work needs to focus on performance optimisations

VŠE PRAHA

Direction of further research

- The method is general but suffers from severe performance issues
  - Even for very small toy problems tens of binary variables
  - Real-world problems computationally infeasible

- Linearization with less binary variables
- Simplification (1 change of shape within criterion)
- Run as many Despotis UTA as there are positions in which the change of shape may occur