Robust ordinal regression for multiple criteria sorting problems within MAUT

Miłosz Kadziński

Poznań University of Technology, Poland, milosz.kadzinski@cs.put.poznan.pl

IDSS Seminar, October 20, 2009
1 Introduction

2 Sorting with a set of value function

3 The most representative value function
   - Principle
   - Three stage-procedure
   - One-stage procedure
   - Illustrative example

4 Conclusions
1 Introduction

2 Sorting with a set of value function
   - Principle
   - Three stage-procedure
   - One-stage procedure
   - Illustrative example

3 The most representative value function

4 Conclusions
Outline

1. Introduction
2. Sorting with a set of value function
   - Principle
   - Three stage-procedure
   - One-stage procedure
   - Illustrative example
3. The most representative value function
4. Conclusions

S. Greco, M. Kadziński, R. Słowiński

Robust ordinal regression for MCSPs within MAUT
Outline

1. Introduction
2. Sorting with a set of value function
3. The most representative value function
   - Principle
   - Three stage-procedure
   - One-stage procedure
   - Illustrative example
4. Conclusions

S. Greco, M. Kadziński, R. Słowiński
Robust ordinal regression for MCSPs within MAUT
Multiple Criteria Sorting Problems

Characteristics

- **Actions** described by evaluation vectors to be assigned to classes
- **Pre-defined ordered** (or unsorted) classes
- Classes have a semantic definition
- Assignment to classes is grounded on absolute evaluation of actions
- **No relative comparisons** is required, ≠ choice, ranking
Multiple Criteria Sorting Problems

Characteristics

- **Actions** described by evaluation vectors to be assigned to classes
- **Pre-defined ordered** (or unsorted) classes
- Classes have a semantic definition
- Assignment to classes is grounded on absolute evaluation of actions
- **No relative comparisons** is required, ≠ choice, ranking

Robust ordinal regression for MCSPs within MAUT
Real-world sorting problems

- Financial management and economics: business failure prediction, credit risk assessment for firms and consumers, country risk assessment
- Environmental and energy management, ecology: analysis of different energy policies
- Human resources management: assignment to appropriate occupation groups, incentive package groups
- Marketing: customer satisfaction measurement, development of market penetration strategies
- Tourism: star-based categories
Definitions and notation

Given data

- $A = \{a_1, a_2, \ldots, a_i, \ldots, a_m\}$ - a finite set of $m$ actions
- $g_1, g_2, \ldots, g_j, \ldots, g_n$ - $n$ evaluation criteria, $g_j : A \rightarrow \mathbb{R}$ for all $j \in G = \{1, 2, \ldots, n\}$
- $X_j = \{x_j \in \mathbb{R} : g_j(a_i) = x_j, a_i \in A\}$ - the set of all different evaluations on $g_j$, $j \in G$
- $x_j^0, x_j^1, \ldots, x_j^{m_j}$ - the ordered values of $X_j$,
  $x_j^k < x_j^{k+1}, k = 0, 1, \ldots, m_j - 1$
- $C_1, C_2, \ldots, C_p$ - $p$ predefined preference ordered classes, where $C_{h+1}$ is preferred to $C_h$, $h = 1, \ldots, p - 1$, moreover, $H = \{1, \ldots, p\}$
Definitions and notation

Preference model

To represent DM’s preferences, we use an additive value function such that:

\[ U(a) = \sum_{j=1}^{n} u_j(g_j(a)), \]

where the marginal value functions \( u_j \) are such that:

\[ u_j(x_j^k) \leq u_j(x_j^{k+1}), \quad k = 0, 1, \ldots, m_j - 1, j \in G \]

To normalize \( U \) so that \( U(a) \in [0, 1], \forall a \in A \), we set:

\[ u_j(x_j^0) = 0, \quad \forall j \in G \text{ and } \sum_{j=1}^{n} u_j(x_j^{m_j}) = 1 \]
Definitions and notations

Preference information

- $A^R \subseteq A$ - a set of reference actions,
- An assignment example is an action $a^* \in A^R$ for which the DM defined a desired assignment $a^* \rightarrow [C_{LDM}(a^*), C_{RDM}(a^*)]$, i.e. to an interval of contiguous classes $C_{LDM}(a^*), C_{LDM}(a^*)+1, \ldots, C_{RDM}(a^*)$
- An assignment example is said to be precise if $L^{DM}(a^*) = R^{DM}(a^*)$, and imprecise, otherwise
Compatible preference information

Given a value function $U$, a set of assignment examples is said to be consistent with $U$ iff:

$$\forall a^*, b^* \in A^R, U(a^*) \geq U(b^*) \Rightarrow R^{DM}(a^*) \geq L^{DM}(b^*),$$

which is equivalent to:

$$\forall a^*, b^* \in A^R, L^{DM}(a^*) > R^{DM}(b^*) \Rightarrow U(a^*) > U(b^*)$$

We will suppose the DM provides a set of assignment examples consistent with $U$
Compatible preference information

- Reference actions: $A^R = \{a_1, a_2, a_3\}$
- Classes: $C_1 << C_2 << C_3 << C_4$
- A set of assignment examples:

  $a_1 \Rightarrow [C_1, C_2], a_2 \Rightarrow [C_2, C_3], a_3 \Rightarrow [C_4, C_4]$

- Resulting constraints:

  $U(a_3) > U(a_1)$
  $U(a_3) > U(a_2)$
**Value driven sorting procedures**

**Definition**

*Value driven* sorting procedures aim to assign each action to one class or a set of contiguous classes, using a value function $U$ in such a way that if $U(a) > U(b)$ then $a$ is assigned to a class not worse than $b$.

**Sorting procedures**

Given a single additive value function $U$, two different sorting procedures can be considered:

- **threshold-based** value driven sorting procedure
- **example-based** value driven sorting procedure
### Threshold-based sorting procedure

- Action $a \in A$ is assigned to class $C_h$, denoted as $a \rightarrow C_h$, iff $U(a) \in [b_{h-1}^U, b_h)$
- Threshold $b_{h-1}$ corresponds to the minimum value for an action $a$ to be assigned to class $C_h$
- Threshold $b_h$ corresponds to the supremum value for an action $a$ to be assigned to class $C_h$
- We impose $b_{h-1} < b_h$, $\forall h \in H$ and we set $b_0 = 0$ and $b_p > 1$
Threshold-based sorting procedure

\[ U(a) = \sum_{i} u_i(g_i(a)) \]

S. Greco, M. Kadziński, R. Słowiński

Robust ordinal regression for MCSPs within MAUT
Value driven sorting procedures

Example-based sorting procedure

- It is driven by a value function $U$ and its associated assignment examples $A^R \subseteq A$
- It assigns an action $a$ to an interval of classes: $[C_{L^U(a)}, C_{R^U(a)}]$

$L^U(a) = \text{Max} \left\{ L^{DM}(a^*) : U(a^*) \leq U(a), a^* \in A^R \right\}$

$R^U(a) = \text{Min} \left\{ R^{DM}(a^*) : U(a^*) \geq U(a), a^* \in A^R \right\}$
Value driven sorting procedures

Example-based sorting procedure

- For each non-reference action $a \in A \setminus A^R$ the indices satisfy the following condition:
  \[ L^U(a) \leq R^U(a) \]
- Each reference action $a^* \in A^R$ is assigned to an interval of classes, such that:
  \[ L^U(a^*) \geq L^{DM}(a^*) \quad R^U(a^*) \leq R^{DM}(a^*) \]
Proposition

Consider the case where \( L^{DM}(a^*) = R^{DM}(a^*), \forall a^* \in A^R \)

Assuming the use of a single value function \( U \) in the example-based sorting procedure, if we choose the threshold \( b_h^U, h = 1, \ldots, p - 1 \) in the interval

\[
\text{Max } a^* \rightarrow c_h \{ U(a^*) \}, \text{Min } a^* \rightarrow c_{h+1} \{ U(a^*) \}
\]

we obtain a **threshold-based sorting procedure** that restores the assignment examples and assigns each non-reference action \( a \in A \setminus A^R \) to a single class in the interval \([C_{LU}(a), C_{RU}(a)]\) stemming from the **example-based sorting procedure**
Threshold-based vs example-based sorting

reference actions assigned to $C_1$

$U$

reference actions assigned to $C_2$

$U$

reference actions assigned to $C_3$

$b_1^U$

$b_2^U$
Consider $b \in A$

$L^U(b) = C_1$, $R^U(b) = C_2$
Threshold-based vs example-based sorting

Proposition

Consider the case where $L_{DM}(a^*) \leq R_{DM}(a^*), \forall a^* \in A^R$

Assuming the use of a single value function $U$ in the example-based sorting procedure, if we choose the threshold $b^U_h, h = 1, \ldots, p - 1$ in the interval

$$\left[ \text{Max } a^* : R_{DM}(a^*) \leq h \{ U(a^*) \}, \text{Min } a^* : L_{DM}(a^*) > h \{ U(a^*) \} \right]$$

with $b^U_h < b^U_{h+1}$,

we obtain a threshold-based sorting procedure that restores the assignment examples and assigns each non-reference action $a \in A \setminus A^R$ to a single class in the interval $[C_{LU}(a), C_{RU}(a)]$ stemming from the example-based sorting procedure.
Threshold-based vs example-based sorting

Assignment of non-reference action

Ranges of thresholds

Assignment examples

U(a)

U(a')
Consider $b \in A$, with $U(b) \in [U(a_6), U(a_7)]$
Consider $b \in A$, with $U(b) \in U(a_6), U(a_7)$
$L^U(b) = C_2$
Consider $b \in A$, with $U(b) \in [U(a_6), U(a_7)]$.

$L^U(b) = C_2$, $R^U(b) = C_4$
Robust ordinal regression

Ordinal regression paradigm

- Comprehensive preferences on a subset of reference actions is known a priori
- Consistent criteria aggregation model is inferred from this information to be applied on the set of all actions

Observations

- The set of all preference models compatible with the stated indirect preference information can be quite large
- Traditionally, only one specific set is used to give a recommendation
- The choice of a single preference model is either arbitrary or left to the DM
Robust ordinal regression

Comment
Marginal values in characteristic points are unknown
Robust ordinal regression

Comment

In fact, they are intervals
Robust ordinal regression

Comment
The area of all compatible marginal value functions
Robust ordinal regression

Comment

In the area, the marginal compatible value functions must be monotone
Robust ordinal regression

Comment

...and not necessarily piecewise-linear
### Aim

- Take into account **all the sets of preference models compatible with the preference information given by the DM**

### Developed methods

- $\text{UTA}^{\text{GMS}}$ and GRIP (choice and ranking problems)
- $\text{UTADIS}^{\text{GMS}}$ (sorting problems)
- Robust ordinal regression applied to Choquet integral
- Robust ordinal regression applied to group decisions
- Interactive and evolutionary multiobjective optimization methodology
- The most representative value function
The UTADIS-GMS method

Questions

- Is action $x \in A$ sorted in the same way by all compatible value functions?
- Is there at least one compatible value function sorting action $x \in A$ to a given class?
The UTADIS-GMS method

Principles

- The preference information is composed of assignment examples on $A^R \subseteq A$
- A value function is called **compatible** if it is able to restore all assignment examples
- In result, one obtains two assignments for any action $a \in A$:
  1. the possible assignment $C_P(a)$:
     
     $C_P(a) = \{ h \in H : \exists U \in \mathcal{U}_{A^R} \text{ for which } h \in [L^U(a), R^U(a)] \}$
  2. the necessary assignment $C_N(a)$:
     
     $C_N(a) = \{ h \in H : \forall U \in \mathcal{U}_{A^R} \text{ it holds } h \in [L^U(a), R^U(a)] \}$

where $L^U(a)$ and $R^U(a)$ are, respectively, the **worst** and the **best class** to which action $a$ is assigned by value function $U$
The UTADIS-GMS method

General additive compatible value function

An additive value function \( U(a) = \sum_{j=1}^{n} u_j(g_j(a)) \) satisfying the following set of constraints:

- **compatibility** with the preference information:
  \[ \forall a^*, b^* \in A^*, L_{DM}^a(a^*) > R_{DM}^b(b^*) \Rightarrow U(a^*) > U(b^*) \]

- **monotonicity** of the family of criteria \( G: \)
  \[ u_i(g_i(a_{\tau_i(j)}) - u_i(g_i(a_{\tau_i(j-1)})) \geq 0, i = 1, \ldots, n, j = 2, \ldots, m. \]
  where \( \tau_i \) is the permutation on the set of indices of actions from \( A \) that reorders them according to the increasing evaluation on criterion \( g_i: \)
  \[ g_i(a_{\tau_i(1)}) \leq g_i(a_{\tau_i(2)}) \leq \ldots \leq g_i(a_{\tau_i(m-1)}) \leq g_i(a_{\tau_i(m)}) \]

- **normalization** to the interval \([0, 1]::\)
  \[ u_i(\alpha_i) = 0, i = 1, \ldots, n, \quad \sum_{i=1}^{n} u_i(\beta_i) = 1 \]
Remarks about linear constraints

Transforming strict inequalities into weak inequalities

\[ x > y \Rightarrow x \geq y + \varepsilon \]

Set of functions satisfying the particular condition

\[ \varepsilon^* = \max \varepsilon \]

\[ x \geq y + \varepsilon \]

constraints defining set of functions

- if \( \varepsilon^* > 0 \) then there exists at least one function satisfying condition \( x > y \) in the defined set of functions,
- if \( \varepsilon^* \leq 0 \) then there is no function satisfying condition \( x > y \) in the defined set of functions.
The set of compatible value function

Verification whether the set of all compatible value functions $\mathcal{U}_{AR}$ is not empty

$$\mathcal{U}_{AR} \neq \emptyset \iff \varepsilon^* > 0$$

where:

$$\varepsilon^* = \max \varepsilon$$

$$U(a^*) \geq U(b^*) + \varepsilon \iff L^{DM}(a^*) > R^{DM}(b^*) \forall a^*, b^* \in A^R$$

$$u_i(g_i(a_{\tau_i(j)})) - u_i(g_i(a_{\tau_i(j-1)})) \geq 0, i = 1, \ldots, n, j = 2, \ldots, m$$

$$u_i(g_i(a_{\tau_i(1)})) \geq 0, u_i(g_i(a_{\tau_i(m)})) \leq u_i(\beta_i), i = 1, \ldots, n$$

$$u_i(\alpha_i) = 0, i = 1, \ldots, n$$

$$\sum_{i=1}^{n} u_i(\beta_i) = 1$$

(E^A_R)
Possible preference relation

**Definition**

\( a \succeq^P b \) means that \( U(a) \geq U(b) \) for at least one compatible value function

\[
a \succeq^P b \iff \varepsilon^* > 0
\]

where:

\[
\varepsilon^* = \max \varepsilon
\]

\[
U(a) \geq U(b)
\]

\[
U(a^*) \geq U(b^*) + \varepsilon \iff L^{DM}(a^*) > R^{DM}(b^*) \}
\]

\[
u_i(g^j_i) - u_i(g^{j-1}_i) \geq 0, \ i = 1, \ldots, n, \ j = 1, \ldots, \omega + 1
\]

\[
u_i(g^0_i) = 0, \ i = 1, \ldots, n
\]

\[
\sum_{i=1}^n u_i(g^{\omega+1}_i) = 1
\]
Necessary preference relation

**Definition**

\( a \succeq^N b \) means that \( U(a) \geq U(b) \) for all compatible value functions

\[
\begin{align*}
a \succeq^N b & \iff \varepsilon_* \leq 0 \\
\varepsilon_* & = \max \varepsilon
\end{align*}
\]

where:

\[
\begin{align*}
U(b) & \geq U(a) + \varepsilon \\
U(a^*) & \geq U(b^*) + \varepsilon \iff L^{DM}(a^*) > R^{DM}(b^*) \} \forall a^*, b^* \in A^R \\
u_i(g^i_j) - u_i(g^i_{j-1}) & \geq 0, i = 1, \ldots, n, j = 1, \ldots, \omega + 1 \\
u_i(g^0_i) & = 0, i = 1, \ldots, n \\
\sum_{i=1}^n u_i(g^\omega+1_i) & = 1
\end{align*}
\]
Possible assignments

- Minimum possible class:

\[ L_P^U(a) = \text{Max} \left\{ L^{DM}(a^*) : \forall U \in U_{AR}, U(a^*) \leq U(a), a^* \in A^R \right\} = \]

\[ = \text{Max} \left\{ L^{DM}(a^*) : a \preceq_a a^*, a^* \in A^R \right\} \]

- Maximum possible class:

\[ R_P^U(a) = \text{Min} \left\{ R^{DM}(a^*) : \forall U \in U_{AR}, U(a^*) \geq U(a), a^* \in A^R \right\} = \]

\[ = \text{Min} \left\{ R^{DM}(a^*) : a^* \succeq_a a, a^* \in A^R \right\} \]

- Assign to each \( a \in A \) its possible assignment

\[ C_P(a) = [L_P^U(a), R_P^U(a)] \]
Necessary assignments

- Potentially minimum necessary class:

\[
L_N^U(a) = \max \left\{ L_{DM}(a^*) : \exists U \in U_{AR} \text{ for which } U(a^*) \leq U(a), a^* \in A^R \right\} = \max \left\{ L_{DM}(a^*) : a \preceq^P a^*, a^* \in A^R \right\}
\]

- Potentially maximum necessary class:

\[
R_N^U(a) = \min \left\{ R_{DM}(a^*) : \exists U \in U_{AR} \text{ for which } U(a^*) \geq U(a), a^* \in A^R \right\} = \min \left\{ R_{DM}(a^*) : a^* \succeq^P a, a^* \in A^R \right\}
\]

- Assign to each \( a \in A \) its necessary assignment which is \( C_N(a) = [L_N^U(a), R_N^U(a)] \) in case \( L_N^U(a) \leq R_N^U(a) \) and \( C_N(a) = \emptyset \) otherwise
### The UTADIS – GMS method

#### Example

- \( A = \{a_1, a_2, a_3, a_4, a_5, a_6\} \)
- \( G = \{g_1, g_2\} \) - a consistent family \( G \) of 2 criteria with an increasing direction of preference
- \( C = \{C_1, C_2, C_3, C_4, C_5\} \) where \( C_{h+1} \gg C_h, h = 1, 2, 3, 4 \)
- Evaluation table:

<table>
<thead>
<tr>
<th></th>
<th>( a_1 )</th>
<th>( a_2 )</th>
<th>( a_3 )</th>
<th>( a_4 )</th>
<th>( a_5 )</th>
<th>( a_6 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( g_1 )</td>
<td>2</td>
<td>6</td>
<td>7</td>
<td>5</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>( g_2 )</td>
<td>3</td>
<td>7</td>
<td>9</td>
<td>3</td>
<td>5</td>
<td>9</td>
</tr>
</tbody>
</table>
Example

Step 1: The DM provides **exemplary assignments** for actions in the reference set $A_R = \{a_1, a_3, a_5\}$. (S)he feels confident that:

$$
\begin{aligned}
a_1 &\rightarrow [C_1, C_1] \\
a_3 &\rightarrow [C_4, C_4] \\
a_5 &\rightarrow [C_2, C_3]
\end{aligned}
$$

Step 2: Provided preference information is consistent and, consequently, a set of compatible value functions is not empty.
The UTADIS – GMS method

Example

Step 1: The DM provides **exemplary assignments** for actions in the reference set $A_R = \{a_1, a_3, a_5\}$. (S)he feels confident that:

\[
\begin{align*}
a_1 & \rightarrow [C_1, C_1] \\
a_3 & \rightarrow [C_4, C_4] \\
a_5 & \rightarrow [C_2, C_3]
\end{align*}
\]

Step 2: Provided preference information is **consistent** and, consequently, a set of compatible value functions is not empty.
The UTADIS – GMS method

### Example

**Step 3: Matrix of relations $\succeq^P$ and $\succeq^N$**

<table>
<thead>
<tr>
<th></th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
<th>$a_4$</th>
<th>$a_5$</th>
<th>$a_6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_1$</td>
<td>T</td>
<td>F</td>
<td>-</td>
<td>T</td>
<td>-</td>
<td>F</td>
</tr>
<tr>
<td>$a_2$</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>-</td>
<td>T</td>
<td>-</td>
</tr>
<tr>
<td>$a_3$</td>
<td>-</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>-</td>
<td>T</td>
</tr>
<tr>
<td>$a_4$</td>
<td>T</td>
<td>-</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>-</td>
</tr>
<tr>
<td>$a_5$</td>
<td>-</td>
<td>T</td>
<td>-</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>$a_6$</td>
<td>T</td>
<td>T</td>
<td>-</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

### Step 4: Calculate boundary indices

\[
\begin{align*}
L^U_P(a_2) & = \max \{ L^M_D(a_1) = C_1, L^M_D(a_5) = C_2 \} = C_2 \\
R^U_P(a_2) & = \min \{ R^M_D(a_3) = C_4 \} = C_4 \\
L^U_N(a_2) & = \max \{ L^M_D(a_1) = C_1, L^M_D(a_3) = C_4, L^M_D(a_5) = C_2 \} = C_4 \\
R^U_N(a_2) & = \min \{ R^M_D(a_3) = C_4, R^M_D(a_5) = C_3 \} = C_3
\end{align*}
\]
The UTADIS – GMS method

Example

Step 3: Matrix of relations $\succeq_P$ and $\succeq_N$

\[
\begin{array}{cccccc|cccccc}
& a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 \\
\hline
a_1 & T & F & - & T & - & F & T & F & - & F & - & F \\
a_2 & T & T & T & - & T & - & T & T & F & - & T & - \\
a_3 & - & T & T & T & - & T & - & T & T & T & - & F \\
a_4 & T & - & F & T & T & - & T & - & F & T & F & - \\
a_5 & - & T & - & T & T & F & - & F & - & T & T & F \\
a_6 & T & - & T & - & T & T & T & - & T & - & T & T \\
\end{array}
\]

Step 4: Calculate boundary indices

\[
\begin{align*}
L^U_P(a_2) &= \text{Max} \{ L^{DM}(a_1) = C_1, L^{DM}(a_5) = C_2 \} = C_2 \\
R^U_P(a_2) &= \text{Min} \{ R^{DM}(a_3) = C_4 \} = C_4 \\
L^U_N(a_2) &= \text{Max} \{ L^{DM}(a_1) = C_1, L^{DM}(a_3) = C_4, L^{DM}(a_5) = C_2 \} = C_4 \\
R^U_N(a_2) &= \text{Min} \{ R^{DM}(a_3) = C_4, R^{DM}(a_5) = C_3 \} = C_3
\end{align*}
\]
The UTADIS-GMS method

Example

Step 5, 6: Matrix of possible and necessary assignments

<table>
<thead>
<tr>
<th></th>
<th>$L_P$</th>
<th>$R_P$</th>
<th>$L_N$</th>
<th>$R_N$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_1$</td>
<td>$C_1$</td>
<td>$C_1$</td>
<td>$C_1$</td>
<td>$C_1$</td>
</tr>
<tr>
<td>$a_2$</td>
<td>$C_2$</td>
<td>$C_4$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$a_3$</td>
<td>$C_4$</td>
<td>$C_4$</td>
<td>$C_4$</td>
<td>$C_4$</td>
</tr>
<tr>
<td>$a_4$</td>
<td>$C_1$</td>
<td>$C_3$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$a_5$</td>
<td>$C_2$</td>
<td>$C_3$</td>
<td>$C_2$</td>
<td>$C_3$</td>
</tr>
<tr>
<td>$a_6$</td>
<td>$C_4$</td>
<td>$C_5$</td>
<td>$C_4$</td>
<td>$C_4$</td>
</tr>
</tbody>
</table>

$L_P^U(a) \leq L_N^U(a)$ and $R_N^U(a) \leq R_P^U(a)$.
The most representative value function

Why to search for the function?

- To see and know the most representative value function among all the compatible ones
- To assess relative importance of the criteria
- To assign a score (value) to the actions
- To work out the most representative assignments
- To identify the most representative function without loosing the advantage of taking into account all compatible value functions
- To exhibit it explicitly along with the results of the UTADIS\textsuperscript{GMS} method
The most representative value function

Question

Which value function is the most representative one in the set of value functions compatible with the preference information?
One for all, all for one

- **One for all**: one value function is representing all compatible value functions
- **All for one**: all compatible value functions contribute to the definition of the most representative value function
The most representative value function in sorting

First idea

1. **Maximize** the minimal difference between values of actions \( a, b \in A \) for which possible assignments are **disjoint**:

\[
[L_P^U(a), R_P^U(a)] \cap [L_P^U(b), R_P^U(b)] = \emptyset
\]

2. **Minimize** the maximal difference between values of actions \( a, b \in A \) for which possible assignments are **not disjoint**:

\[
[L_P^U(a), R_P^U(a)] \cap [L_P^U(b), R_P^U(b)] \neq \emptyset
\]

Comment

Wrong
The most representative value function in sorting

First idea

1. **Maximize** the minimal difference between values of actions $a, b \in A$ for which possible assignments are **disjoint**:

$$[L_P^U(a), R_P^U(a)] \cap [L_P^U(b), R_P^U(b)] = \emptyset$$

2. **Minimize** the maximal difference between values of actions $a, b \in A$ for which possible assignments are **not disjoint**:

$$[L_P^U(a), R_P^U(a)] \cap [L_P^U(b), R_P^U(b)] \neq \emptyset$$

Comment

Wrong
Why first idea is not correct?

\[ \begin{align*}
\exists U_1 & \in U_{A^r} & & C_{U_1}^U(b) \\
\exists U_2 & \in U_{A^r} & & C_{U_2}^U(b) \\
\exists U_3 & \in U_{A^r} & & C_{U_3}^U(b)
\end{align*} \]
The most representative value function in sorting

Final idea

1. **Maximize** the minimal difference between values of actions \(a, b \in A\) for which **possible assignments are disjoint**

2. **Maximize** the minimal difference between values of actions \(a, b \in A\) for which:
   - for all value function \(U\) \(a\) is assigned to a class **not worse** than the class of \(b\),
   - and, for at least one compatible value function \(a\) is assigned to a class which is **better** than the class of \(b\)

3. **Minimize** the maximal difference between values of actions \(a, b \in A\)
   - being in the **same class** for all compatible value functions \(U\)
   - or, for which the order of classes is **not univocal**
The most representative value function in sorting

Possible assignments for actions $a, b \in A$ are disjoint:

$$[L^U_P(a), R^U_P(a)] \cap [L^U_P(b), R^U_P(b)] = \emptyset.$$  

The difference between values of actions should be as large as possible.
Three-stage procedure

First stage

1. For all pairs of actions \((a, b)\), such that \(L_P^U(a) > R_P^U(b)\), add the following constraint to the linear programming constraints of UTADIS\(^{GMS}\):

\[
U(a) \geq U(b) + \varepsilon.
\]

2. Maximize \(\varepsilon\).

3. Add the constraint \(\varepsilon = \varepsilon^*\), with \(\varepsilon^* = \max \varepsilon\) from point 2), to the linear programming constraints of point 1).
The most representative value function in sorting

Condition in the second stage

\[ a \rightarrow b \iff \forall U \in \mathcal{U}_{A^R} : (L_U^U(a) \geq R_U^U(b)) \]

and

\[ (\exists U \in \mathcal{U}_{A^R} : L_U^U(a) > R_U^U(b)); \]

The difference between values of actions should be as large as possible.
Three-stage procedure

Second stage

1. For all pairs of actions \((a, b)\), such that \(a \succeq b\), add the following constraint to the linear programming constraints of point 3) of first stage:

\[ U(a) \geq U(b) + \gamma. \]

2. Maximize \(\gamma\).

3. Add the constraint \(\gamma = \gamma^*\), with \(\gamma^* = \max \gamma\) from point 2), to the linear programming constraints of point 1).
Introduction
Sorting with a set of value function
The most representative value function
Conclusions

Principle
Three stage-procedure
One-stage procedure
Illustrative example

The most representative value function in sorting

\[ a \sim b \iff (\forall U \in U_{AR} : L^U(a) = L^U(b) \text{ and } R^U(a) = R^U(b)) \]
\[ \text{or } (\exists U_1, U_2 \in U_{AR} : L^{U_1}(a) > R^{U_1}(b) \text{ and } L^{U_2}(b) > R^{U_2}(a)) \]

Condition in the third stage

The difference between values of actions should be as small as possible.

S. Greco, M. Kadziński, R. Słowiński
Robust ordinal regression for MCSPs within MAUT
Three-stage procedure

Third stage

1. For all pairs of actions \((a, b)\), such that \(a \sim b\) add the following constraints to the linear programming constraints of point 3) of second stage:

\[ U(a) - U(b) \leq \delta \text{ and } U(b) - U(a) \leq \delta. \]

2. Minimize \(\delta\).
1 Introduction

2 Sorting with a set of value function

3 The most representative value function
   - Principle
   - Three stage-procedure
   - One-stage procedure
   - Illustrative example

4 Conclusions
One-stage procedure

1. For all pairs of actions \((a, b)\), such that \(L_P^U(a) > R_P^U(b)\) add:
   \[ U(a) \geq U(b) + \varepsilon. \]
   to the linear programming constraints of UTADIS\(^{GMS}\).

2. For all pairs of actions \((a, b)\), such that \(a \succ \succ b\) add:
   \[ U(a) \geq U(b) + \gamma. \]
   to the linear programming constraints from point 1).

3. For all pairs of actions \((a, b)\), such that \(a \sim \succ b\) add:
   \[ U(a) - U(b) \leq \delta \text{ and } U(b) - U(a) \leq \delta. \]
   to the linear programming constraints from point 2).

4. Maximize \(M\varepsilon + N\gamma - \delta\)
   where \(M >> N >> 1\).
Outline

1. Introduction
2. Sorting with a set of value function
3. The most representative value function
   - Principle
   - Three stage-procedure
   - One-stage procedure
   - Illustrative example
4. Conclusions
Global MBA classes

Problem statement and given data

- Actions: 30 MBA programs offered by different universities (originally from the Financial Times; reconsidered by Koksalan et al., 2008)
- Criteria: 20 factors grouped under three main criteria: alumni career progress, diversity and idea generation
- Classes: 5 preference-ordered
- Evaluation table

<table>
<thead>
<tr>
<th>Program name</th>
<th>Alumni career progress</th>
<th>Diversity idea generation</th>
<th>Idea generation</th>
</tr>
</thead>
<tbody>
<tr>
<td>London Business</td>
<td>68.78</td>
<td>62.03</td>
<td>59.87</td>
</tr>
<tr>
<td>Yale University</td>
<td>79.01</td>
<td>25.98</td>
<td>51.84</td>
</tr>
<tr>
<td>Carnegie Mellon</td>
<td>54.02</td>
<td>18.69</td>
<td>71.93</td>
</tr>
<tr>
<td>Duke University</td>
<td>64.05</td>
<td>27.25</td>
<td>64.68</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
The UTADIS – GMS method

Example

Step 1: Ask the DM for possibly imprecise sorting examples. The DM provides exemplary assignments for 8 actions in the reference set:

- London Business School → 5
- University of North Carolina → 3
- University of Maryland → 1

...  

Step 2: Verify whether the set of compatible value functions is not empty. Provided preference information is consistent.
The UTADIS – GMS method

Example

Step 1: Ask the DM for possibly imprecise sorting examples. The DM provides exemplary assignments for 8 actions in the reference set:

- London Business School → 5
- University of North Carolina → 3
- University of Maryland → 1
- ...

Step 2: Verify whether the set of compatible value functions is not empty. Provided preference information is consistent.
The UTADIS – GMS method

Example

Step 3: Determine the possible sorting $C_P(a)$ for each considered action $a \in A$.

Step 4: For all pairs of actions $(a, b)$, such that $L_P^U(a) > R_P^U(b)$, add the following constraint to the linear programming constraints of UTADIS$^{GMS}$:

$$U(a) \geq U(b) + \varepsilon.$$

There are 72 pairs of actions $(a_1, b_1)$ for which $L_P^U(a_1) > R_P^U(b_1)$.
Example

Step 3: Determine the possible sorting $C_P(a)$ for each considered action $a \in A$.

Step 4: For all pairs of actions $(a, b)$, such that $L^U_P(a) > R^U_P(b)$, add the following constraint to the linear programming constraints of UTADIS$^{\text{GMS}}$:

$$U(a) \geq U(b) + \varepsilon.$$  

There are 72 pairs of actions $(a_1, b_1)$ for which $L^U_P(a_1) > R^U_P(b_1)$.
Example

Step 5: Determine the relation $\succ \rightarrow$ for all pairs of actions $(a, b)$ with $a, b \in A$.

Step 6: For all pairs of actions $(a, b)$, such that $a \succ \rightarrow b$, add the following constraint to the linear programming constraints from step 4):

$$U(a) \geq U(b) + \gamma.$$

There are 181 pairs of actions $(a_2, b_2)$ for which $a_2 \succ \rightarrow b_2$. 
The UTADIS – GMS method

Example

Step 5: Determine the relation $\succ\rightarrow$ for all pairs of actions $(a, b)$ with $a, b \in A$.

Step 6: For all pairs of actions $(a, b)$, such that $a \succ\rightarrow b$, add the following constraint to the linear programming constraints from step 4):

$$U(a) \geq U(b) + \gamma.$$ 

There are 181 pairs of actions $(a_2, b_2)$ for which $a_2 \succ\rightarrow b_2$. 
The UTADIS – GMS method

Example

Step 7: Determine the relation $\sim \rightarrow$ for all pairs of actions (a,b) with $a, b \in A$.

Step 8: For all pairs of actions $(a, b)$, such that $a \sim \rightarrow b$ add the following constraints to the linear programming constraints from step 6):

$$U(a) - U(b) \leq \delta \text{ and } U(b) - U(a) \leq \delta.$$ 

There are 251 pairs of actions $(a_3, b_3)$ for which $a_3 \sim \rightarrow b_3$. 

S. Greco, M. Kadziński, R. Słowiński

Robust ordinal regression for MCSPs within MAUT
Example

Step 7: Determine the relation $\sim \Rightarrow$ for all pairs of actions $(a,b)$ with $a, b \in A$.

Step 8: For all pairs of actions $(a, b)$, such that $a \sim \Rightarrow b$ add the following constraints to the linear programming constraints from step 6):

$$U(a) - U(b) \leq \delta \text{ and } U(b) - U(a) \leq \delta.$$ 

There are 251 pairs of actions $(a_3, b_3)$ for which $a_3 \sim \Rightarrow b_3$. 

S. Greco, M. Kadziński, R. Słowiński

Robust ordinal regression for MCSPs within MAUT
The UTADIS – GMS method

Example

Step 9: Maximize $M\varepsilon + N\gamma - \delta$

$$U(a) \geq U(b) + \varepsilon \Leftrightarrow L^U_P(a) > R^U_P(b) \quad \forall a, b \in A$$

$$U(a) \geq U(b) + \delta \Leftrightarrow a \succ b \quad \forall a, b \in A$$

$$U(a) - U(b) \leq \gamma \Leftrightarrow a \sim b \quad \forall a, b \in A$$

$$U(b) - U(a) \leq \gamma \Leftrightarrow a \sim b \quad \forall a, b \in A$$

$$U(a^*) \geq U(b^*) + \varepsilon \Leftrightarrow L^{DM}(a^*) > R^{DM}(b^*) \quad \forall a^*, b^* \in A^R$$

$$u_i(g_i(a_{\tau_i(j)})) - u_i(g_i(a_{\tau_i(j-1)})) \geq 0, \quad i = 1, \ldots, n, \quad j = 2, \ldots, m$$

$$u_i(g_i(a_{\tau_i(1)})) \geq 0, \quad u_i(g_i(a_{\tau_i(m)})) \leq u_i(\beta_i), \quad i = 1, \ldots, n$$

$$u_i(\alpha_i) = 0, \quad i = 1, \ldots, n$$

$$\sum_{i=1}^n u_i(\beta_i) = 1$$

where $M$ and $N$ are arbitrarily large positive constant such that $M > N > 1$. 
The most representative value function in sorting

Results of optimization

- \( \varepsilon = 0.2 \), e.g. \( \{ U(LBS) = 0.857, U(Yale) = 0.657 \} \),
  \( \{ U(LBS) = 0.857, U(Maryland) = 0.057 \} \),
- \( \gamma = 0.029 \), e.g. \( \{ U(LBS) = 0.857, U(\text{Pennsylv.}) = 0.828 \} \),
  \( \{ U(LBS) = 0.857, U(Maryland) = 0.057 \} \),
- \( \delta = 0.714 \), e.g. \( \{ U(\text{Pennsylv.}) = 0.114, U(W. Ontario) = 0.114 \} \),
  \( \{ U(Rotterdam) = 0.828, U(W. Ontario) = 0.114 \} \).
The most representative value function in sorting

Characteristics

- The characteristic points correspond to the evaluation values of the considered actions.
- The constructed functions are usually not strictly monotonic, which results from the form of optimized function $M \varepsilon + N \gamma - \delta$.
- Although in the figure connections between characteristic points are linear, it would be sufficient if they reflected the monotonic character.
Example

Step 10: Optionally, conduct an example-based sorting procedure driven by the value function from point 9 and assignment examples from point 1 in order to determine the most representative assignment for each action in the considered set.
**Decision Deck**

**Aim of the platform**

Decision Deck (D2) aims to develop a tool to support decision makers in evaluating actions in a multiple criteria and multiple experts context.
Technical aspects

- Current implementation of plugin works on the second version of Decision Deck platform (1.1)
- The plugin is an OSGI bundle - dynamically loadable collection of classes, resources, and configuration files
- Data access is achieved through Hibernate
- To analyze potential inconsistency and verify the truth of preference relations it uses GLKP linear solver
- To visualize relations and assignments of actions in form of tables it uses standard Java classes and for visualisation of the most representative value function it uses JChart
Functionality of the UTADIS-GMS plugin

Alternatives
- add / remove / edit

Criteria
- add / remove / edit

Classes
- add / remove / edit / reorganize order

Reference set
- add / remove alternatives

Evaluation matrix
- fill / edit

Preference information
- add / remove / edit / solve inconsistency

Possible and necessary preference relations
- calculate / visualize

Possible and necessary assignments
- calculate / visualize

The most representative value function
- calculate / visualize
Summary

- New approach to multicriteria sorting of actions
- Preference information is used within a robust regression approach to build a complete set of compatible additive value functions
- Identification of possible and necessary consequences of provided information
- The most representative value function built on relations defined on the whole set of value functions
- Separate method (“most representative results”) or complementary use along with UTADIS$^{GMS}$
Future works

1. The most representative preference model:
   - The most representative value function for group decisions
   - The most representative set of parameters for outranking methods

2. Decision Deck:
   - ELECTRE-GMS and the most representative set of parameters on Decision Desktop
   - GRIP and UTADIS-GMS on diviz