Preference elicitation for MCDA
Robust elicitation of sorting model

Vincent Mousseau

[1]LAMSADE, Université Paris-Dauphine, France
mousseau@lamsade.dauphine.fr

International Doctoral School, Troina, Italy

COST action "Algorithmic Decision Theory"

April 2008

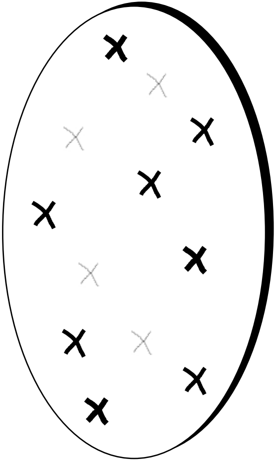
Contents

Contents

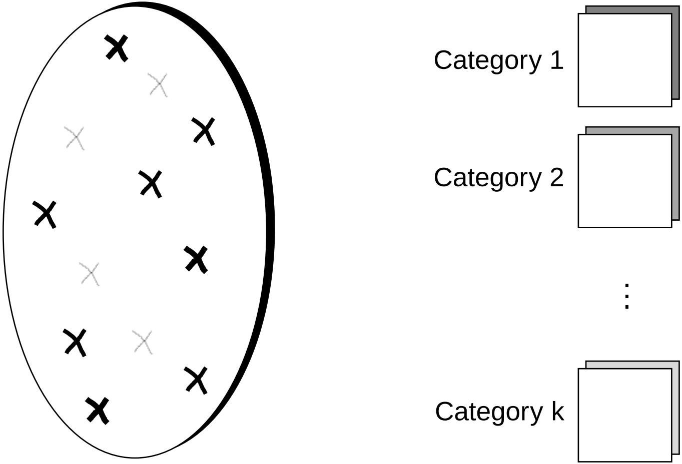
Problem statements

- Assign alternatives to pre-defined categories

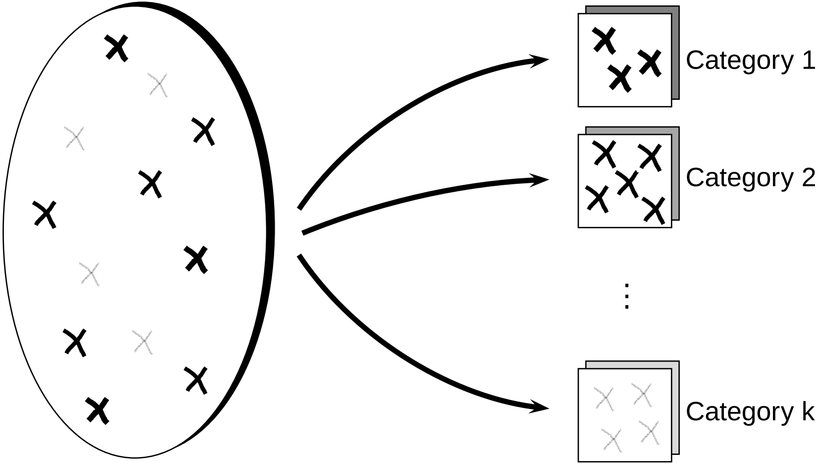
Sorting problem statement



Sorting problem statement



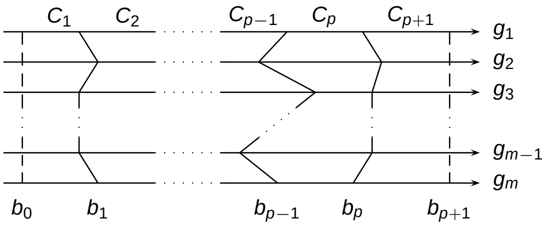
Sorting problem statement



Contents

Electre Tri method

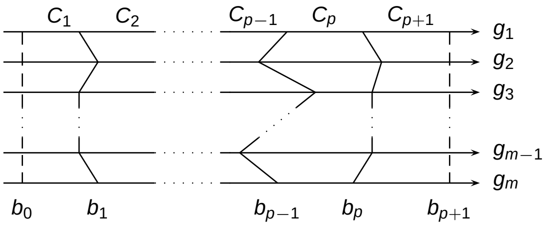
1. Define categories using limit profiles $B = \{b_1, b_2, \ldots, b_p\}$,



2. Compare a to $b_1, b_2, ..., b_p$ using an outranking relation S.
3. Assign a to a category C_h according to how a compares to $b_h, h = 1..p$.

Electre Tri method

1. Define categories using limit profiles $B = \{b_1, b_2, \ldots, b_p\}$,

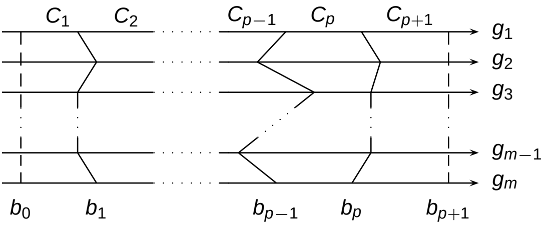


2. Compare a to $b_1, b_2, ..., b_p$ using an outranking relation S.

3. Assign a to a category C_h according to how a compares to $b_h, h = 1..p$.

Electre Tri method

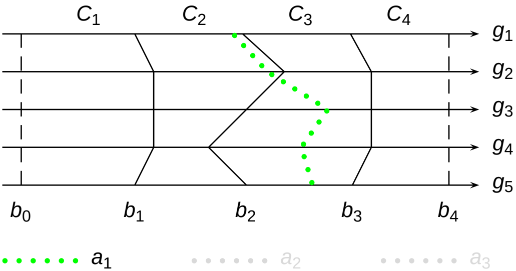
1. Define categories using limit profiles $B = \{b_1, b_2, \ldots, b_p\}$,



2. Compare a to $b_1, b_2, ..., b_p$ using an outranking relation S.

3. Assign a to a category C_h according to how a compares to $b_h, h = 1..p$.

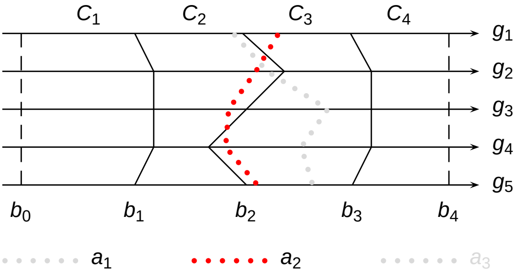
Electre Tri method



	b_0	b_1	b_2	b_3	b_4
a_1	a_1Pb_0	a_1Pb_1	a_1Pb_2	b_3Pa_1	b_4Pa_1
a_2					
a_3					

- $Pes(a_1) = C_3$,
- $Opt(a_1) = C_3$,

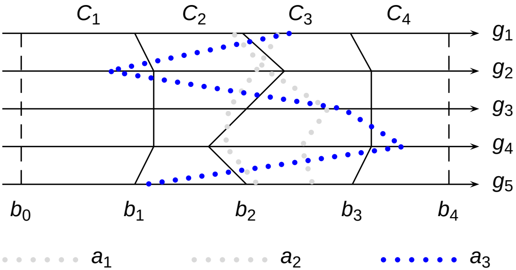
Electre Tri method



	b_0	b_1	b_2	b_3	b_4
a_1	a_1Pb_0	a_1Pb_1	a_1Pb_2	b_3Pa_1	b_4Pa_1
a_2	a_2Pb_0	a_2Pb_1	a_2Ib_2	b_3Pa_2	b_4Pa_2
a_3					

- $Pes(a_1) = C_3$, $Pes(a_2) = C_3$,
- $Opt(a_1) = C_3$, $Opt(a_2) = C_3$,

Electre Tri method



	b_0	b_1	b_2	b_3	b_4
a_1	a_1Pb_0	a_1Pb_1	a_1Pb_2	b_3Pa_1	b_4Pa_1
a_2	a_2Pb_0	a_2Pb_1	a_2Ib_2	b_3Pa_2	b_4Pa_2
a_3	a_3Pb_0	a_3Pb_1	a_3Rb_2	b_3Pa_3	b_4Pa_3

- $Pes(a_1) = C_3$, $Pes(a_2) = C_3$ and $Pes(a_3) = C_2$,
- $Opt(a_1) = C_3$, $Opt(a_2) = C_3$ and $Opt(a_3) = C_3$,

Electre Tri method

- **Pseudo-conjunctive procedure (pessimistic)** :

  a) Compare a successively to b_i, for i=p,p-1, ..., 0,
  b) Consider b_h the first profile such that aSb_h,
  Assign a to category C_{h+1}.

- **Pseudo-disjonctive procedure (optimistic)** :

  a) Compare a successively to b_i, i=1, 2, ..., $p+1$,
  b) Consider b_h the first profile b_h such that $b_h \succ a$,
  Assign a to category C_h.

- If $Pes(a)$ ($Opt(a)$, resp.) is the assignment category a with the pessimistic procedure (optimistic resp.), it holds:
  - $Pes(a) \leq Opt(a)$
  - $Pes(a) < Opt(a)$ iff a is incomparable to at least one profile.

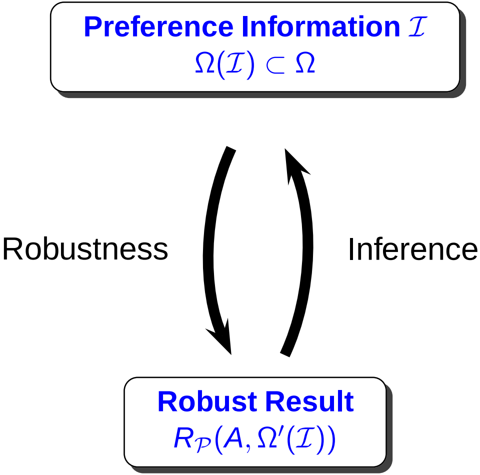
Electre Tri method

- In Electre Tri pessimistic procedure,
  $a \longrightarrow C_h$ iff $a \succsim b_{h-1}$ and $\neg(a \succsim b_h)$
- In Electre Tri optimistic procedure,
  $a \longrightarrow C_h$ iff $\neg(b_{h-1} \succ a)$ and $b_h \succ b_h$

Contents

Elicitation and robustness

- "Exact knowledge" of $\omega \in \Omega \;\rightarrow\; R_{\mathcal{P}}(A, \omega)$,
- "Incomplete knowledge" on $\Omega(\mathcal{I}) \subset \Omega \;\rightarrow\; R_{\mathcal{P}}(A, \Omega(\mathcal{I}))$,
- $R_{\mathcal{P}}(A, \Omega')$ is the result de $\mathcal{P}$ applied to A considering the "incomplete knowledge"on $\Omega(\mathcal{I}) \subset \Omega$,
- Computing $R_{\mathcal{P}}(A, \Omega(\mathcal{I}))$ require to develop specific algorithms ([Dias, Climaco 2000], [Greco, Mousseau, Slowinski 2007])


Preference Information $\mathcal{I}$
$\Omega(\mathcal{I}) \subset \Omega$
Robustness
Inference
Robust Result
$R_{\mathcal{P}}(A, \Omega'(\mathcal{I}))$

Computing robust assignments

- [Dias, Clímaco 2000] propose algorithms to compute robust assignments,
- grounded on the computation of the interval in which $\sigma(a, b_h),\ a \in A,\ h \in B$ vary knowing $\mathcal{I}$,
- Principle : identify $max(a, \Omega(\mathcal{I}))$ ($min(a, \Omega(\mathcal{I}))$, resp.) the index of the best (worst, resp.) category to which a can be assigned considering $\Omega(\mathcal{I})$,

Computing robust assignments

- $Min_{\omega \in \Omega(\mathcal{I})} \sigma(a, b_1) \geq \lambda \;\Rightarrow\; \neg(a \rightarrow C_1)$,
- $Min_{\omega \in \Omega(\mathcal{I})} \sigma(a, b_2) \geq \lambda \;\Rightarrow\; \neg(a \rightarrow C_2)$,
- ...

- $Max_{\omega \in \Omega(\mathcal{I})} \sigma_d(a, b_p) < \lambda \;\Rightarrow\; \neg(a \rightarrow C_p)$,
- $Max_{\omega \in \Omega(\mathcal{I})} \sigma_d(a, b_{p-1}) < \lambda \;\Rightarrow\; \neg(a \rightarrow C_{p-1})$,
- ...

$\longrightarrow$ Hence we can determine that $a \rightarrow [C_{min}, C_{max}]$

Computing robust assignments

```
Begin
    h ← p   (best category)
    While  ∃ω ∈ Ω(I) : ¬(aS_ω b_{h−1})
    Do
        h ← h-1
    End While
    min(a, Ω(I)) ← h
End
```

$$\exists\omega \in \Omega(\mathcal{I}) : \neg(aS_\omega b_{h-1}) \qquad min(a,\Omega(\mathcal{I})) \leftarrow h$$

```
Begin
    h ← p   (best category)
    While  ¬(aS_ω b_{h−1}), ∀ω ∈ Ω(I)
    Do
        h ← h-1
    End While
    max(a, Ω(I)) ← h
End
```

$$\neg(aS_\omega b_{h-1}),\ \forall\omega \in \Omega(\mathcal{I}) \qquad max(a,\Omega(\mathcal{I})) \leftarrow h$$

Contents

Infer a preference model

- Inference procedure = algorithm that, starting from an information $\mathcal{I}$ identifies $\omega^*(\mathcal{I})$ which "best match" $\mathcal{I}$ when using $\mathcal{P}$,
- An inference procedure is grounded on the resolution of a mathematical program:
  - decision variables= parameters to infer,
  - objective fonction = minimize an "error" fonction (how good $\mathcal{I}$ is accounted for),
  - constraints = way by which $\mathcal{I}$ is expressed in terms of the preference parameters of $\mathcal{P}$.

Disaggregation

Preference Information $\mathcal{I}$

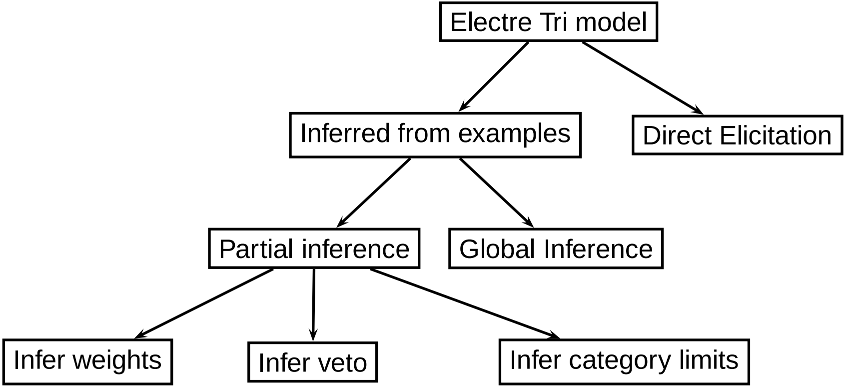
↓

Inference procedure

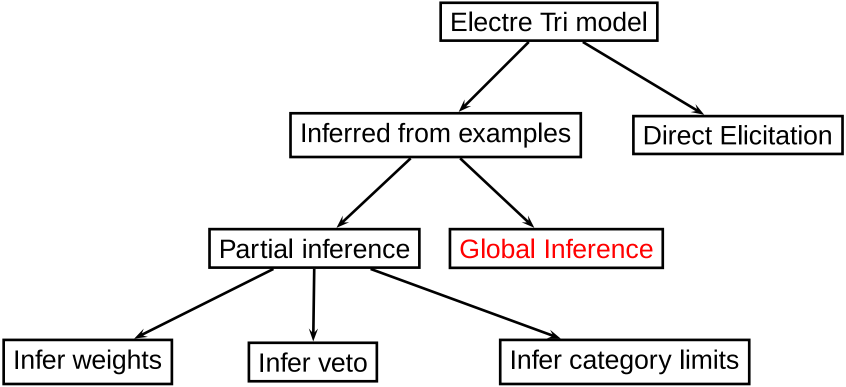
↓

inferred parameters: $\omega^*(\mathcal{I})$
($\mathcal{P}$**,** $\omega^*(\mathcal{I})$**) = preference model that "best" match** $\mathcal{I}$

Inference of an Electre Tri model



Inference of an Electre Tri model



Global inference

- Consider the assignment example $a \rightarrow_{DM} C_h$, $C_h = [b_{h-1}, b_h[$,
- With Electre Tri pessimistic rule $a \rightarrow C_h \Leftrightarrow aSb_{h-1}$ and $\neg aSb_h$
  i.e. $\sigma(a, b_{h-1}) \geq \lambda$ and $\sigma(a, b_h) < \lambda$,
- x_a and y_a are slack variables defined as:
  $\sigma(a, b_{h-1}) - x_a = \lambda$ and $\sigma(a, b_h) + y_a + \varepsilon = \lambda$, ($\varepsilon$ small positive value)
- If $x_a \geq 0$ and $y_a \geq 0$, then $a \rightarrow C_h$, $\forall \lambda' \in [\lambda - y_a, \lambda + x_a]$,
- Consider A^* a set of alternatives for which the DM expresses a desired assignment,
- If $x_a \geq 0$ and $y_a \geq 0$ $\forall a \in A^*$, then Electre Tri restores assignment examples in A^* properly.

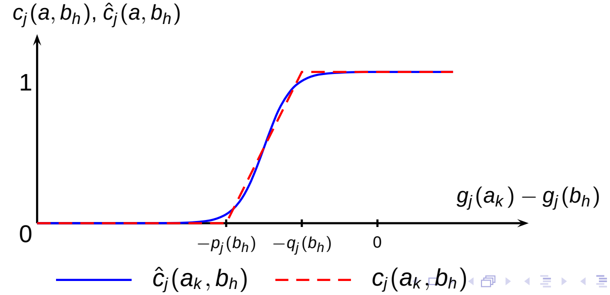
Global Inference

$$
\begin{aligned}
& Max\ \alpha \\
& s.t. \quad \alpha \leq x_a, \quad \forall a \in A^* && (1)\\
& \alpha \leq y_a, \quad \forall a \in A^* && (2)\\
& \sigma(a, b_{h_a-1}) - x_k = \lambda, \quad \forall a \in A^* && (3)\\
& \sigma(a, b_{h_a}) + y_k + \varepsilon = \lambda, \quad \forall a \in A^* && (4)\\
& \lambda \in [0.5, 1] && (5)\\
& g_j(b_{h+1}) \geq g_j(b_h) + p_j(b_h) + p_j(b_{h+1}), \forall j \in F, \forall h \in B && (6)\\
& v_j(b_h) \geq p_j(b_h) \geq q_j(b_h), \quad \forall j \in F, \forall h \in B && (7)\\
& k_j \geq 0, q_j(b_h) \geq 0, \quad \forall j \in F, \forall h \in B && (8)\\
& \sum_{j \in F} k_j = 1 && (9)\\
& \text{all positive variables but } \alpha,\ x_a,\ y_a,\ \forall a \in A^* && (10)
\end{aligned}
$$

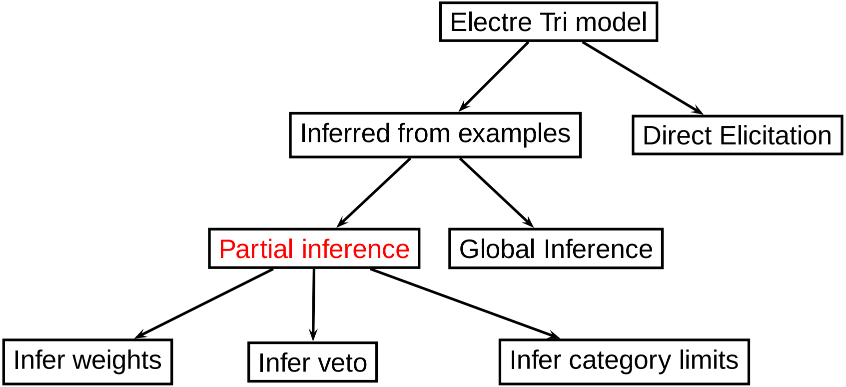
Global Inference

$$\sigma(a,b_h) = C(a,b_h) \times \prod_{j \in \overline{F}} \frac{1-d_j(a,b_h)}{1-C(a,b_h)} \text{ où}$$
$$\overline{F} = \{j : d_j(a,b_h) > C(a,b_h)\}$$

$$\hat{c}_j(a,b_h) = \frac{1}{1+\exp\left[\frac{-5.55}{p_j(b_h)-q_j(b_h)} \cdot \left(g_j(a)-g_j(b_h)+\frac{p_j(b_h)+q_j(b_h)}{2}\right)\right]}$$



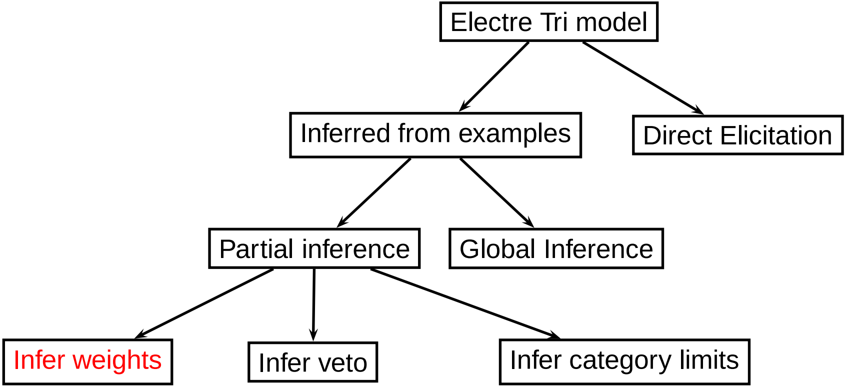
Inference of an Electre Tri model



Partial inference

- The inference of all parameters leads to to large mathematical program for real world problems,
- To circumvent this difficulty, it is possible to sequentially solve programs which infer a subset of parameters,
- Problem : optimal value of inferred parameters correspond to values that best match assignment examples the other parameters being fixed.

Inference of an Electre Tri model

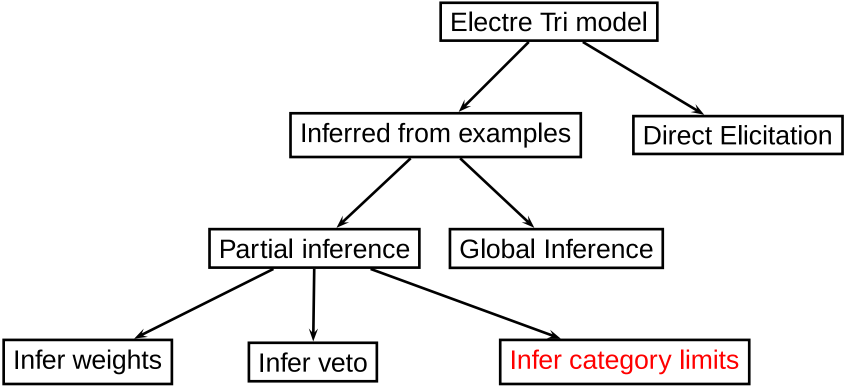


Inference of k_j and λ

- if we infer k_j and λ only then inference lead to a linear program,

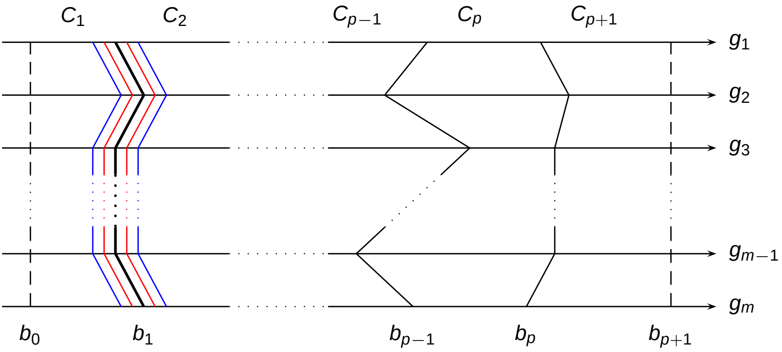
$$
\begin{array}{ll}
Max\ \alpha & \\
\quad s.t. & \alpha \leq x_a, \quad \forall a \in A^* \\
& \alpha \leq y_a, \quad \forall a \in A^* \\
& \sum_{j \in F} k_j c_j(a, b_{h_a - 1}) \; - \; x_k = \lambda, \quad \forall a \in A^* \\
& \sum_{j \in F} k_j c_j(a, b_{h_a}) \; + \; y_k \; + \; \varepsilon \; = \; \lambda, \quad \forall a \in A^* \\
& \lambda \in [0.5, 1], \; k_j \geq 0, \; \forall j \in F, \; \sum_{j \in F} k_j = 1 \\
& \text{all variables positive but } \alpha, \; x_a, \; y_a, \; \forall a \in A^*
\end{array}
$$

Inferring an Electre Tri model



Inference of category limits

- Infer values for $g_j(b_h)$, $q_j(b_h)$ and $p_j(b_h)$, $\forall h \in B$, $\forall j \in F$ (the other parameters' values being fixed),



Inference of category limits

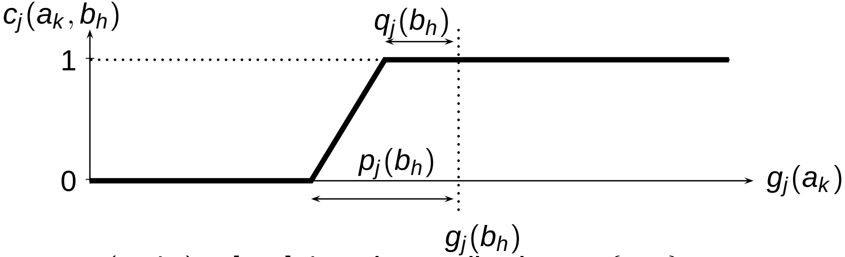
- It is difficult to infer directly the values for $g_j(b_h)$, $q_j(b_h)$ and $p_j(b_h)$, $\forall h \in B$, $\forall j \in F$ (values for k_j and $v_j(b_h)$ being fixed),
- 2 successives phases:

Phase 1 : infer how alternatives compare with category limits, i.e., partial concordance indices $c_j(a, b_h)$ and $c_j(b_h, a)$ that best match assignment example,

Phase 2 : determine values for $g_j(b_h)$, $q_j(b_h)$ and $p_j(b_h)$, compatible with partial concordance indices obtained in phase 1.

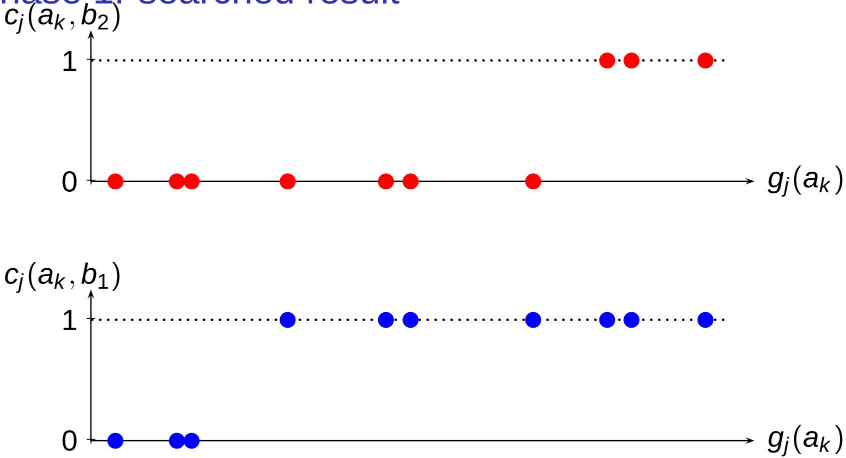
Phase 1

infer how alternatives should compare to profiles (partial concordance indices $c_j(a, b_h)$ and $c_j(b_h, a)$) so that assignment examples are “best” accounted for.



- $c_j(a_i, b_h) \in [0, 1]$, but almost all values $\in \{0, 1\}$,
- Variables are $c_j(a_i, b_h)$, λ (majority level) and a slack variable β

Phase 1: searched result



Phase 1

$$\max \beta$$

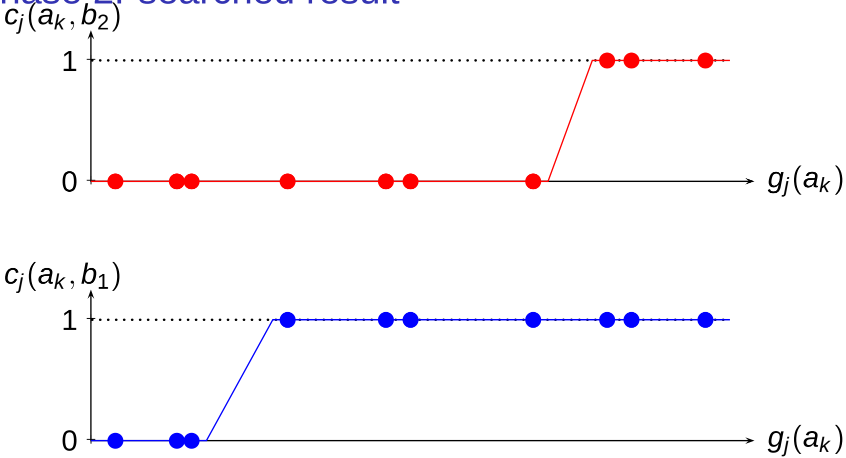
$$
\text{s.t.} \quad
\begin{array}{rcl}
\beta & \leq & \sum_{j\in F} k_j c_j(a, b_{h_a-1}) - \lambda, \forall a \in A^* \\
\beta + \epsilon & \leq & \lambda - \sum_{j\in F} k_j c_j(a, b_{h_a}), \forall a \in A^* \\
\beta + \epsilon & \leq & \lambda - \sum_{j\in F} k_j c_j(b_{h_a-2}, a), \forall a \in A^* \\
1 & \leq & c_j(a, b_h) + c_j(b_h, a), \forall j \in F, \forall a \in A^*, h \in B \\
c_j(a, b_{h+1}) & \leq & c_j(a, b_h), \forall j \in F, \forall a \in A^*, h = 1, 2, ..., p-1 \\
c_j(b_{h+1}, a) & \geq & c_j(b_h, a), \forall j \in F, \forall a \in A^*, h = 1, 2, ..., p-1 \\
c_j(a, b_h) & \leq & c_j(a', b_h), \forall j \in F, \forall a, a' \in A^*, h \in B, \text{ if } g_j(a) < g_j(a') \\
c_j(a, b_h) & = & c_j(a', b_h), \forall j \in F, \forall a, a' \in A^*, h \in B, \text{ if } g_j(a) = g_j(a') \\
c_j(b_h, a) & \geq & c_j(b_h, a'), \forall j \in F, \forall a, a' \in A^*, h \in B, \text{ if } g_j(a) < g_j(a') \\
c_j(b_h, a) & = & c_j(b_h, a'), \forall j \in F, \forall a, a' \in A^*, h \in B, \text{ if } g_j(a) = g_j(a') \\
0.5 & \leq & \lambda \leq 1 \\
c_j(a, b_h) & \in & \{0, 1\}, \forall j \in F, \forall a \in A^*, h \in B \\
c_j(b_h, a) & \in & \{0, 1\}, \forall j \in F, \forall a \in A^*, h \in B
\end{array}
$$

Phase 2

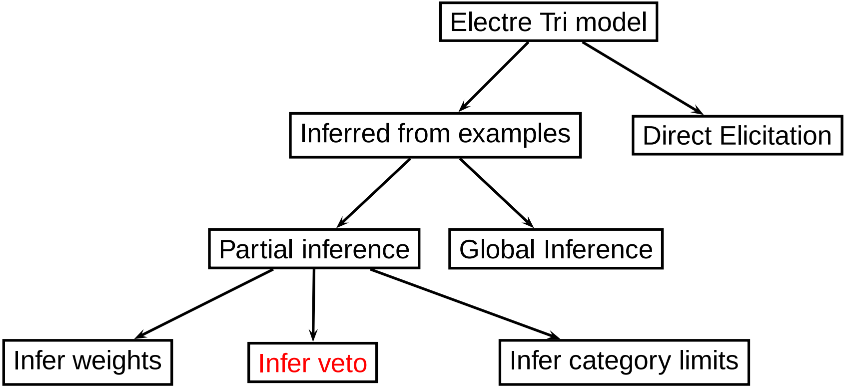
Once $c_j(a_k, b_h)$ and $c_j(b_h, a_k)$ are computed, any values for $g_j(b_h), p_j(b_h)$ and $q_j(b_h)$ verifying the following conditions are acceptable:

- $c_j(a_k, b_h) = 0 \Rightarrow g_j(b_h) - p_j(b_h) \geq g_j(a_k)$
- $c_j(a_k, b_h) = 1 \Rightarrow g_j(b_h) - q_j(b_h) \leq g_j(a_k)$
- $c_j(b_h, a_k) = 0 \Rightarrow g_j(b_h) + p_j(b_h) \leq g_j(a_k)$
- $c_j(b_h, a_k) = 1 \Rightarrow g_j(b_h) + q_j(b_h) \geq g_j(a_k)$
- $g_j(b_{h+1}) \geq g_j(b_h)$
- $p_j(b_h) \geq q_j(b_h) \geq 0$

Phase 2: searched result



Inference of an Electre Tri model



Inference of vetos

- Infer the veto thresholds $v_j(b_h)$ from assignment examples, the value of the other parameters being fixed,
- We distinguish cases where:
  - one single veto threshold is inferred,
  - several veto thresholds are inferred,

Inference of a single veto

- All preference parameters are fixed except v_i (supposed constant),
- Assignment examples induce constraints:

$$\left\{\begin{array}{rcl} \sigma(a,b_h) & \geq & \lambda, \quad \forall a, b_h \text{ t.q. } Cat_{min}(a) = h+1 \\ \sigma(a,b_h) & \leq & \lambda+\varepsilon, \quad \forall a, b_h \text{ t.q. } Cat_{max}(a) = h \quad \text{or} \\ v_i & \geq & p_i + \varepsilon \end{array}\right.$$

$$\text{or } \sigma(a,b_h) = C(a,b_h).\prod_{j\in \overline{F}\setminus\{i\}}((1-d_j(a,b_h))).(1-d_i(a,b_h)) = K_i(a,b_h).(1-d_i(a,b_h))$$

- Consider the relation S_{-i}, $aS_{-i}b_h$ means a outranks b_h in absence of veto on g_i, i.e., aSb_h is possible for some values for v_i,

$$aS_{-i}b_h \Leftrightarrow K_i(a,b_h) \geq \lambda \Leftrightarrow (d_j(a,b_h) = 0 \Rightarrow aSb_h)$$

Inference of a single veto

- Consider the constraint of the form $\sigma(a, b_h) \geq \lambda$,
  - if $\neg aS_{-i}b_h$ then it is not possible to find a value for v_i (inconsistent information),
  - if $C(a, b_h) = 1$ then any value for v_i will make the constraint true (redundant information)
- Consider a constraint of the form $\sigma(a, b_h) \leq \lambda + \varepsilon$,
  - if $C(a, b_h) = 1$ then it is impossible to find a value for v_i (inconsistent information),
  - if $\neg aS_{-i}b_h$ then any value for v_i will make the constraint true (redundant information)

Contents

Inconsistency management

- Consider the ELECTRE TRI method for which the DM is not able to assign precise values for k_j and λ,
- Each assignment example induce 2 linear constraints on weights and λ,
- $\rightarrow$ Polyhedron of acceptable values for k_j and λ
- When the preference information can not be represented in the ELECTRE TRI model, the polyhedron of admissible values for k_j and λ empty
  $\rightarrow$ inconsistency

Inconsistency management

- Assignment example define m constraints

$$\left\{ \begin{array}{lll} \sum_{j=1}^{n} \alpha_{1j} w_j + \alpha'_1 \lambda & \geq & \beta_1 \\ & \vdots & \\ \sum_{j=1}^{n} \alpha_{(m-1)j} w_j + \alpha'_{m-1} \lambda & \geq & \beta_{m-1} \\ \sum_{j=1}^{n} \alpha_{mj} w_j + \alpha'_m \lambda & \geq & \beta_m \end{array} \right. \quad [1]$$

- Denote $I = \{1, \ldots, m\}$; $S \subset I$ solves [1] iff $I \setminus S \neq \emptyset$
- We look for $S_1, S_2, \ldots, S_p \subset I$ such that:
  - (i) S_i solves [1], $i \in \{1, 2, ..., p\}$;
  - (ii) $S_i \not\subseteq S_j, i, j \in \{1, ..., p\}, i \neq j$;
  - (iii) $|S_i| \leq |S_j|, i, j \in \{1, 2, ..., p\}, i < j$;
  - (iv) if $\exists$ S solves [1] s.t. $S \not\subseteq S_i, \ \forall i = 1, 2, \ldots, p$, then $|S| > |S_p|$.

Inconsistency management

- Consider y_i ($\in \{0,1\}$, $i \in I$), s.t. :

$$y_i = 1 \text{ if constraint } i \text{ is deleted}$$
$$= 0 \text{ otherwise}$$

$$P_1 \begin{cases} Min & \sum_{i \in I} y_i \\ \text{s.t.} & \sum_{j=1}^{n} \alpha_{ij} x_j + \alpha_i' \lambda + M y_i \geq \beta_i, \quad \forall i \in I \\ & x_j \geq 0, \quad j = 1, \dots, n \\ & y_i \in \{0,1\}, \quad \forall i \in I \end{cases}$$

- $S_1 = \{i \in I : y_i^* = 1\}$ corresponds to a (or several) subset(s) of constraints solving [1] of smaller cardinality,
- We define P_2 adding to P_1 the constraint $\sum_{i \in S_1} y_i \leq |S_1| - 1$

Inconsistency management

- P_{k+1} is defined adding to P_k the constraint $\sum_{i \in S_k} y_i \leq |S_k| - 1$
- $S_1, S_2, \ldots, S_k$ are computed, and the algorithm stops when $|S_{k+1}| > \Omega$ (or when no more solution exists),

```
Begin
     k ⟵ 1
     moresol ⟵ true
     While moresol
       Solve PM_k
       If (PM_k has no solution) or (PM_k has an optimal value > Ω)
         Then moresol ⟵ false
         Else
           - S_k ⟵ {i ∈ I : y_i^* = 1}
           - Add constraint ∑_{i∈S_k} y_i ≤ |S_k| − 1 to PM_k → define PM_{k+1}
           - k ⟵k+1
       End if
     End while
End
```

Inconsistency management

- Each S_i corresponds to a set of assignment example (presented to the DM),
- S_i sets represent “incompatibles” assignment examples, each of them specify a way to solve inconsistency.

Contents

IRIS v2.0: Software illustration

- IRIS implements robust elicitation of an Electre Tri model whinthin a constructive learning perspective,
- In IRIS learning concerns k_j and λ,
- IRIS determines robust assignements,
- IRIS detects inconsistencies and proposes alternative solutions to restore consistency,

IRIS v2.0: Software illustration

- Data required by IRIS as input:
  - category limits ($g_j(b_h), q_j(b_h)$ and $p_j(b_h)$),
  - veto thresholds ($v_j(b_h)$),
  - assignment examples (possibly imprecise)
  - additional contraints on k_j and λ.

IRIS v2.0: Software illustration

Output information computed by IRIS in the **absence of inconsistency** :

- a central weight vector that best match the provided information,
- For each alternative:
  - its assignment when using the “central” weight vector,
  - robust assignment, *i.e.*, $[C_{min}(a), C_{max}(a)]$
  - for each $C_h \in [C_{min}(a), C_{max}(a)]$, weights that lead to the assignment,

IRIS v2.0: Software illustration

Information fournie par IRIS in **presence of inconsistency** :

- a central weight vector that best match the provided information,
- For each alternative, its assignment when using the "central" weight vector (even if it differs from the required assignment),
- a list of minimal subsets of constraints, that if deleted lead to a consistent model.

IRIS v2.0: Software illustration

- Strategy for use:
  - accounting for a large number of assignment examples,
  - progressive integration of assignment examples,

IRIS v2.0 : Example

Assigning students evaluated on 5 dimensions to 4 categories
$\longrightarrow$ refusal, hesitating refusal, hesitating acceptance, acceptance.

	Crit 1	Crit 2	Crit 3	Crit 4	Crit 5
a_0	2	4	7	16	11
a_1	5	4	7	11	11
a_2	7	9	11	10	16
a_3	8	5	11	10	3
a_4	10	11	11	6	3
a_5	10	4	12	5	14
a_6	11	17	18	16	9
a_7	11	16	16	11	15
a_8	12	4	3	5	17
a_9	13	8	15	7	6
a_{10}	14	10	16	7	6
a_{11}	15	10	1	5	17
a_{12}	15	10	11	18	8
a_{13}	18	10	1	15	8
a_{14}	19	16	16	11	8

IRIS v2.0 : Example

progressive integration of assignment examples,

- $a_8 \rightarrow [C_3, C_4]$,
- $a_{14} \rightarrow C_4$,
- $a_5 \rightarrow [C_1, C_2]$,
- $k_4 \geq 0.01$
- $k_1 \geq 0.33$
- $a_7 \rightarrow [C_1, C_2]$ (inconsistency),

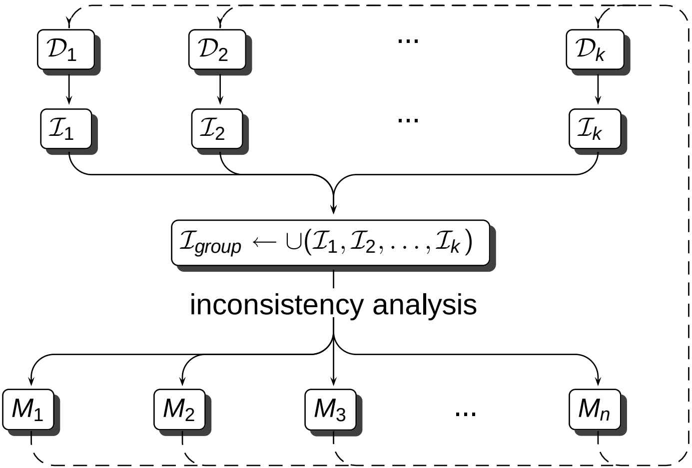
IRIS v2.0 : Example

accounting for a large number of assignment examples,

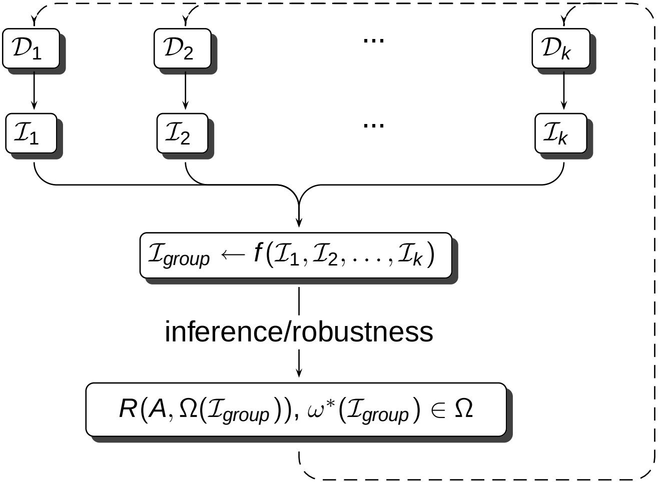
- $a_0 \longrightarrow C_1$,
- $a_1 \longrightarrow [C_3, C_4]$ (error judgment),
- $a_2 \longrightarrow C_3$,
- $a_3 \longrightarrow C_1$,
- $a_6 \longrightarrow [C_1, C_2]$,
- $a_{10} \longrightarrow [C_3, C_4]$,
- $a_{12} \longrightarrow C_4$

Contents

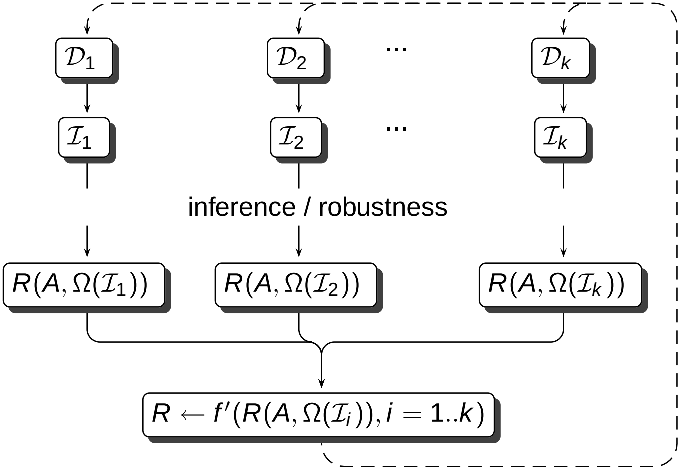
Multiple DMs paradigms



Multiple DMs paradigms



Multiple DMs paradigms



Contents

Proposed methodology

In the proposed methodology:

- DMs agree on the evaluation criteria,
- DMs consider the same set A and evaluation table,
- DMs agree on the definition of categories, thus on limit profiles,
- DMs interact on assignment examples,
- Aggregation/disaggregation principles support interaction,
- DMs refine the information iteratively.

Two main difficulties arise:

- Possible disagreement on assignment examples among DMs,
- Finding an agreement on assignment examples that is consistent.

Proposed methodology

In the proposed methodology:

- DMs agree on the evaluation criteria,
- DMs consider the same set A and evaluation table,
- DMs agree on the definition of categories, thus on limit profiles,
- DMs interact on assignment examples,
- Aggregation/disaggregation principles support interaction,
- DMs refine the information iteratively.

Two main difficulties arise:

- Possible disagreement on assignment examples among DMs,
- Finding an agreement on assignment examples that is consistent.

Proposed methodology

In the proposed methodology:

- DMs agree on the evaluation criteria,
- DMs consider the same set A and evaluation table,
- DMs agree on the definition of categories, thus on limit profiles,
- DMs interact on assignment examples,
- Aggregation/disaggregation principles support interaction,
- DMs refine the information iteratively.

Two main difficulties arise:

- Possible disagreement on assignment examples among DMs,
- Finding an agreement on assignment examples that is consistent.

Proposed methodology

In the proposed methodology:

- DMs agree on the evaluation criteria,
- DMs consider the same set A and evaluation table,
- DMs agree on the definition of categories, thus on limit profiles,
- DMs interact on assignment examples,
- Aggregation/disaggregation principles support interaction,
- DMs refine the information iteratively.

Two main difficulties arise:

- Possible disagreement on assignment examples among DMs,
- Finding an agreement on assignment examples that is consistent.

Proposed methodology

In the proposed methodology:

- DMs agree on the evaluation criteria,
- DMs consider the same set A and evaluation table,
- DMs agree on the definition of categories, thus on limit profiles,
- DMs interact on assignment examples,
- Aggregation/disaggregation principles support interaction,
- DMs refine the information iteratively.

Two main difficulties arise:

- Possible disagreement on assignment examples among DMs,
- Finding an agreement on assignment examples that is consistent.

Proposed methodology

In the proposed methodology:

- DMs agree on the evaluation criteria,
- DMs consider the same set A and evaluation table,
- DMs agree on the definition of categories, thus on limit profiles,
- DMs interact on assignment examples,
- Aggregation/disaggregation principles support interaction,
- DMs refine the information iteratively.

Two main difficulties arise:

- Possible disagreement on assignment examples among DMs,
- Finding an agreement on assignment examples that is consistent.

Proposed methodology

In the proposed methodology:

- DMs agree on the evaluation criteria,
- DMs consider the same set A and evaluation table,
- DMs agree on the definition of categories, thus on limit profiles,
- DMs interact on assignment examples,
- Aggregation/disaggregation principles support interaction,
- DMs refine the information iteratively.

Two main difficulties arise:

- Possible disagreement on assignment examples among DMs,
- Finding an agreement on assignment examples that is consistent.

Proposed methodology

In the proposed methodology:

- DMs agree on the evaluation criteria,
- DMs consider the same set A and evaluation table,
- DMs agree on the definition of categories, thus on limit profiles,
- DMs interact on assignment examples,
- Aggregation/disaggregation principles support interaction,
- DMs refine the information iteratively.

Two main difficulties arise:

- Possible disagreement on assignment examples among DMs,
- Finding an agreement on assignment examples that is consistent.

Proposed methodology

The proposed methodology accounts for these two issues:

- The necessity to make DMs converge toward a collective set of robust assignments and finally a common set of inferred parameters,
- The necessity to make DMs being and staying collectively as well as individually consistent.

Proposed methodology

The proposed methodology accounts for these two issues:

- The necessity to make DMs converge toward a collective set of robust assignments and finally a common set of inferred parameters,
- The necessity to make DMs being and staying collectively as well as individually consistent.

Proposed methodology

- Two level are identified :
  - k individual models
  - 1 collective model
- Each individual model is defined by:
  - a set of assignment examples $\mathcal{I}$,
  - the corresponding $\Omega(\mathcal{I})$, $R(A, \Omega(\mathcal{I}))$ and $\omega^*(\mathcal{I}) \in \Omega(\mathcal{I})$,
- Each DM starts with an individual (consistent) model,
- In the iterative process, the collective model is build progressively by integrating assignment examples,
- At each iteration, each individual model should be consistent and compatible with the collective model.

Proposed methodology

- Two level are identified :
  - k individual models
  - 1 collective model
- Each individual model is defined by:
  - a set of assignment examples $\mathcal{I}$,
  - the corresponding $\Omega(\mathcal{I})$, $R(A, \Omega(\mathcal{I}))$ and $\omega^*(\mathcal{I}) \in \Omega(\mathcal{I})$,
- Each DM starts with an individual (consistent) model,
- In the iterative process, the collective model is build progressively by integrating assignment examples,
- At each iteration, each individual model should be consistent and compatible with the collective model.

Proposed methodology

- Two level are identified :
  - k individual models
  - 1 collective model
- Each individual model is defined by:
  - a set of assignment examples $\mathcal{I}$,
  - the corresponding $\Omega(\mathcal{I})$, $R(A, \Omega(\mathcal{I}))$ and $\omega^*(\mathcal{I}) \in \Omega(\mathcal{I})$,
- Each DM starts with an individual (consistent) model,
- In the iterative process, the collective model is build progressively by integrating assignment examples,
- At each iteration, each individual model should be consistent and compatible with the collective model.

Proposed methodology

- Two level are identified :
  - k individual models
  - 1 collective model
- Each individual model is defined by:
  - a set of assignment examples $\mathcal{I}$,
  - the corresponding $\Omega(\mathcal{I})$, $R(A, \Omega(\mathcal{I}))$ and $\omega^*(\mathcal{I}) \in \Omega(\mathcal{I})$,
- Each DM starts with an individual (consistent) model,
- In the iterative process, the collective model is build progressively by integrating assignment examples,
- At each iteration, each individual model should be consistent and compatible with the collective model.

Proposed methodology

- Two level are identified :
  - k individual models
  - 1 collective model
- Each individual model is defined by:
  - a set of assignment examples $\mathcal{I}$,
  - the corresponding $\Omega(\mathcal{I})$, $R(A, \Omega(\mathcal{I}))$ and $\omega^*(\mathcal{I}) \in \Omega(\mathcal{I})$,
- Each DM starts with an individual (consistent) model,
- In the iterative process, the collective model is build progressively by integrating assignment examples,
- At each iteration, each individual model should be consistent and compatible with the collective model.

Proposed methodology

- *Step 1*:
  - Each DM defines a consistent set of assign. examples
  - The collective model has no assignment example
    $(Min(a_i) = C_1, Max(a_i) = C_n)$
- *Step 2*: DMs discuss in order to agree on an assign. example
- *Step 3*: The agreed assignment example is incorporated in the collective model and in each individual model (each DM may privately revise inputs by deleting/modifying examples). New robust assignments are computed for each DM.
- *Step 4*: If the collective model is satisfactory or no further agreement can be found, then Stop, else go to step 2

Proposed methodology

- *Step 1*:
  - Each DM defines a consistent set of assign. examples
  - The collective model has no assignment example

    $(Min(a_i) = C_1, Max(a_i) = C_n)$

- *Step 2*: DMs discuss in order to agree on an assign. example
- *Step 3*: The agreed assignment example is incorporated in the collective model and in each individual model (each DM may privately revise inputs by deleting/modifying examples). New robust assignments are computed for each DM.
- *Step 4*: If the collective model is satisfactory or no further agreement can be found, then Stop, else go to step 2

Proposed methodology

- *Step 1*:
  - Each DM defines a consistent set of assign. examples
  - The collective model has no assignment example

    $(Min(a_i) = C_1, Max(a_i) = C_n)$
- *Step 2*: DMs discuss in order to agree on an assign. example
- *Step 3*: The agreed assignment example is incorporated in the collective model and in each individual model (each DM may privately revise inputs by deleting/modifying examples). New robust assignments are computed for each DM.
- *Step 4*: If the collective model is satisfactory or no further agreement can be found, then Stop, else go to step 2

Proposed methodology

- *Step 1*:
  - Each DM defines a consistent set of assign. examples
  - The collective model has no assignment example

    $(Min(a_i) = C_1, Max(a_i) = C_n)$
- *Step 2*: DMs discuss in order to agree on an assign. example
- *Step 3*: The agreed assignment example is incorporated in the collective model and in each individual model (each DM may privately revise inputs by deleting/modifying examples). New robust assignments are computed for each DM.
- *Step 4*: If the collective model is satisfactory or no further agreement can be found, then Stop, else go to step 2.

Proposed methodology

- Initially in the collective model $C_{min}(a_i) = C_1$, $C_{max}(a_i) = C_n$, $\forall a_i$) and the procedure aims, at each iteration, at narrowing the possible assignments of alternatives,
- A consensus on an assignment example a_i introduces constraints on the parameter values...
- ... which constrain the interval of possible assignments $[C_{min}(a_j), C_{max}(a_j)]$ for $a_j \neq a_i$
- The process stops when each alternative is assigned to a single category or further consensus is difficult to reach.

Proposed methodology

- Initially in the collective model $C_{min}(a_i) = C_1$, $C_{max}(a_i) = C_n$, $\forall a_i$) and the procedure aims, at each iteration, at narrowing the possible assignments of alternatives,
- A consensus on an assignment example a_i introduces constraints on the parameter values...
- ... which constrain the interval of possible assignments $[C_{min}(a_j), C_{max}(a_j)]$ for $a_j \neq a_i$
- The process stops when each alternative is assigned to a single category or further consensus is difficult to reach.

Proposed methodology

- Initially in the collective model $C_{min}(a_i) = C_1$, $C_{max}(a_i) = C_n$, $\forall a_i$) and the procedure aims, at each iteration, at narrowing the possible assignments of alternatives,
- A consensus on an assignment example a_i introduces constraints on the parameter values...
- ... which constrain the interval of possible assignments $[C_{min}(a_j), C_{max}(a_j)]$ for $a_j \neq a_i$
- The process stops when each alternative is assigned to a single category or further consensus is difficult to reach.

Proposed methodology

- Initially in the collective model $C_{min}(a_i) = C_1$, $C_{max}(a_i) = C_n$, $\forall a_i$) and the procedure aims, at each iteration, at narrowing the possible assignments of alternatives,
- A consensus on an assignment example a_i introduces constraints on the parameter values...
- ... which constrain the interval of possible assignments $[C_{min}(a_j), C_{max}(a_j)]$ for $a_j \neq a_i$
- The process stops when each alternative is assigned to a single category or further consensus is difficult to reach.

Contents

Individual/Collective consistency

How to make each individual model consistent with the new assignment example ?

- Suppose all DMs state $a_i \rightarrow C_1$ except DM_1 $a_i \rightarrow C_2$,
- DM_1 can make a concession $a_i \rightarrow C_1$ if he/she accept all consequences in his/her individual model on all assignment ranges:
  - $a_i \rightarrow C_1$ can narrow the assignment range of some other alternatives
  - $a_i \rightarrow C_1$ can contradict an assignment example of the DM_1's private model

Individual/Collective consistency

How to make each individual model consistent with the new assignment example ?

- Suppose all DMs state $a_i \to C_1$ except DM_1 $a_i \to C_2$,
- DM_1 can make a concession $a_i \to C_1$ if he/she accept all consequences in his/her individual model on all assignment ranges:
  - $a_i \to C_1$ can narrow the assignment range of some other alternatives
  - $a_i \to C_1$ can contradict an assignment example of the DM_1's private model

Individual/Collective consistency

How to make each individual model consistent with the new assignment example ?

- Suppose all DMs state $a_i \rightarrow C_1$ except DM_1 $a_i \rightarrow C_2$,
- DM_1 can make a concession $a_i \rightarrow C_1$ if he/she accept all consequences in his/her individual model on all assignment ranges:
  - $a_i \rightarrow C_1$ can narrow the assignment range of some other alternatives
  - $a_i \rightarrow C_1$ can contradict an assignment example of the DM_1's private model

Individual/Collective consistency

How to make each individual model consistent with the new assignment example ?

- Suppose all DMs state $a_i \rightarrow C_1$ except DM_1 $a_i \rightarrow C_2$,
- DM_1 can make a concession $a_i \rightarrow C_1$ if he/she accept all consequences in his/her individual model on all assignment ranges:
  - $a_i \rightarrow C_1$ can narrow the assignment range of some other alternatives
  - $a_i \rightarrow C_1$ can contradict an assignment example of the DM_1's private model

Contents

Choice of a new assignment example

How to choose, at each iteration, a new assignment example?

- $E_k(a_i, C_x) = 1$ if $C_x \in [C^k_{min}(a_i), C^k_{max}(a_i)]$
  $= 0$ otherwise
- $E(a_i, C_x) = \frac{\sum_{k=1}^{K} E_k(a_i, C_x)}{K}$, majority level for $a_i \rightarrow C_x$
- number of "shifts": changing from $a_i \rightarrow C_1$ to $a_i \rightarrow C_3$ is stronger than changing from $a_i \rightarrow C_1$ to $a_i \rightarrow C_2$)

Choice of a new assignment example

How to choose, at each iteration, a new assignment example?

- $E_k(a_i, C_x) = 1$ if $C_x \in [C^k_{min}(a_i), C^k_{max}(a_i)]$
  $= 0$ otherwise
- $E(a_i, C_x) = \frac{\sum_{k=1}^{K} E_k(a_i, C_x)}{K}$, majority level for $a_i \rightarrow C_x$
- number of "shifts": changing from $a_i \rightarrow C_1$ to $a_i \rightarrow C_3$ is stronger than changing from $a_i \rightarrow C_1$ to $a_i \rightarrow C_2$)

Choice of a new assignment example

How to choose, at each iteration, a new assignment example?

- $E_k(a_i, C_x) = 1$ if $C_x \in [C^k_{min}(a_i), C^k_{max}(a_i)]$
  $= 0$ otherwise
- $E(a_i, C_x) = \frac{\sum_{k=1}^{K} E_k(a_i, C_x)}{K}$, majority level for $a_i \rightarrow C_x$
- number of "shifts": changing from $a_i \rightarrow C_1$ to $a_i \rightarrow C_3$ is stronger than changing from $a_i \rightarrow C_1$ to $a_i \rightarrow C_2$)

Choice of a new assignment example

How to choose, at each iteration, a new assignment example?

- $E_k(a_i, C_x) = 1$ if $C_x \in [C^k_{min}(a_i), C^k_{max}(a_i)]$
  $= 0$ otherwise
- $E(a_i, C_x) = \frac{\sum_{k=1}^{K} E_k(a_i, C_x)}{K}$, majority level for $a_i \rightarrow C_x$
- number of "shifts": changing from $a_i \rightarrow C_1$ to $a_i \rightarrow C_3$ is stronger than changing from $a_i \rightarrow C_1$ to $a_i \rightarrow C_2$)

Contents

A short illustrative example

Problem considered :

- Sorting candidates to master degree admission into 4 categories,
- 15 candidates evaluated on 5 criteria, C_1, C_2, C_3 and C_4,
- 4 DMs wish to build a common sorting model,

A short illustrative example

Problem considered :

- Sorting candidates to master degree admission into 4 categories,
- 15 candidates evaluated on 5 criteria, C_1, C_2, C_3 and C_4,
- 4 DMs wish to build a common sorting model,

A short illustrative example

Problem considered :

- Sorting candidates to master degree admission into 4 categories,
- 15 candidates evaluated on 5 criteria, C_1, C_2, C_3 and C_4,
- 4 DMs wish to build a common sorting model,

A short illustrative example

	$g_1(a_i)$	$g_2(a_i)$	$g_3(a_i)$	$g_4(a_i)$	$g_5(a_i)$
a_0	2	4	7	16	11
a_1	5	4	7	11	11
a_2	7	9	11	10	16
a_3	8	5	11	10	3
a_4	10	11	11	6	3
a_5	10	4	12	5	14
a_6	11	17	18	16	9
a_7	11	16	16	11	15
a_8	12	4	3	5	17
a_9	13	8	15	7	6
a_{10}	14	10	16	7	6
a_{11}	15	10	1	5	1
a_{12}	15	10	11	18	8
a_{13}	18	10	1	15	8
a_{14}	19	16	16	11	8

A short illustrative example

	DM_1				DM_2				DM_3				DM_4			
	C_1	C_2	C_3	C_4	C_1	C_2	C_3	C_4	C_1	C_2	C_3	C_4	C_1	C_2	C_3	C_4
a_0	1	0	0	0	0	1	0	0	0	0	1	0	1	1	0	0
a_1	1	1	0	0	0	1	0	0	0	0	1	0	1	1	0	0
a_2	0	1	0	0	0	0	1	0	0	0	1	0	0	1	1	0
a_3	1	1	0	0	0	0	1	0	1	1	0	0	1	1	1	0
a_4	0	0	1	0	0	0	1	0	0	1	0	0	0	1	1	0
a_5	1	1	0	0	0	0	1	0	0	0	1	0	1	0	0	0
a_6	0	0	1	0	0	0	0	1	0	0	1	0	0	0	1	1
a_7	0	0	1	0	0	0	0	1	0	0	1	1	0	0	1	0
a_8	1	1	0	0	1	0	0	0	0	0	1	0	1	0	0	0
a_9	0	1	0	0	0	0	0	1	0	1	0	0	0	1	0	0
a_{10}	0	0	1	0	0	0	0	1	0	1	0	0	0	1	1	0
a_{11}	0	0	1	0	1	0	0	0	1	1	0	0	1	0	0	0
a_{12}	0	0	1	1	0	0	1	0	0	0	1	1	0	0	1	0
a_{13}	0	0	1	1	1	0	0	0	0	0	1	1	1	1	0	0
a_{14}	0	0	0	1	0	0	0	1	0	0	1	0	0	0	1	1

A short illustrative example

	C_1	C_2	C_3	C_4
a_0	50%	50%	25%	0
a_1	50%	75%	25%	0
a_2	0	50%	75%	0
a_3	75%	75%	50%	0
a_4	0	50	75%	0
a_5	50%	50%	50%	0
a_6	0	0	75%	50%
a_7	0	0	75%	50%
a_8	75%	25%	25%	0
a_9	0	75%	25%	25%
a_{10}	0	50%	50%	25%
a_{11}	75%	25%	25%	0
a_{12}	0	0	100%	50%
a_{13}	50%	50%	50%	50%
a_{14}	0	0	50%	75%

A short illustrative example

- All DMs agree that $a_{12} \rightarrow C_3$.
- Two of them agree to change from ($a_{12} \rightarrow C_3$ or C_4) to ($a_{12} \rightarrow C_3$),
- ... and the consequences on their private model.

A short illustrative example

- All DMs agree that $a_{12} \rightarrow C_3$.
- Two of them agree to change from ($a_{12} \rightarrow C_3$ or C_4) to ($a_{12} \rightarrow C_3$),
- ... and the consequences on their private model.

A short illustrative example

- All DMs agree that $a_{12} \longrightarrow C_3$.
- Two of them agree to change from ($a_{12} \longrightarrow C_3$ or C_4) to ($a_{12} \longrightarrow C_3$),
- ... and the consequences on their private model.

A short illustrative example

	C_1	C_2	C_3	C_4
a_0	50%	50%	25%	0
a_1	50%	75%	25%	0
a_2	0	50%	75%	0
a_3	75%	75%	50%	0
a_4	0	50%	75%	0
a_5	50%	50%	50%	0
a_6	0	0	75%	50%
a_7	0	0	75%	50%
a_8	75%	25%	25%	0
a_9	0	75%	25%	25%
a_{10}	0	50%	50%	25%
a_{11}	75%	25%	25%	0
a_{12}	0	0	100%	0
a_{13}	50%	50%	50%	50%
a_{14}	0	0	50%	75%

Contents

Conclusion

- Constructive elicitation of a robust ELECTRE TRI sorting model,
- Account for multiple DMs setting,
- Other elicitation tools need to be designed with respect to MCAPs,
- Plenty of work is to be done to design such elicitation tools.
- Software implementations,