Learning the parameters of a multiple criteria sorting method from large sets of assignemnt examples

Olivier Sobrie[1,2] - Vincent Mousseau[1] - Marc Pirlot[2]

[1]École Centrale de Paris - Laboratoire de Génie Industriel
[2]University of Mons - Faculty of engineering

November 15, 2012

ELECTRE TRI

Purpose of ELECTRE TRI

- Procedure that assign each alternative to a category ;
- Alternative are described by a performance vector ;
- Categories are pre-defined and ordered.

Example of use

- Given a set of m students $A = a_1, ..., a_m$
- Evaluated on n criteria $g_1, ..., g_n$
- Each student a_i is characterized by his performances $g_{i,j}$ on the n criteria
- Assign a grade to each student (satis bene $\prec$ cum laude $\prec$ magna cum laude $\prec$ summa cum laude)

ELECTRE TRI

Different version of ELECTRE TRI

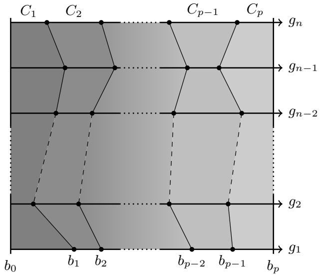
- Originally developed by [Yu, 1992]
- Other variants : ELECTRE TRI-C, ELECTRE TRI-nC, ...
- **MR-Sort : Axiomatic version** (based on [Bouyssou and Marchant, 2007a, Bouyssou and Marchant, 2007b])

ELECTRE TRI

Different version of ELECTRE TRI

- Originally developed by [Yu, 1992]
- Other variants : ELECTRE TRI-C, ELECTRE TRI-nC, ...
- **MR-Sort : Axiomatic version** (based on [Bouyssou and Marchant, 2007a, Bouyssou and Marchant, 2007b])

Parameters



- Profiles' performances ($b_{h,j}$ for $h = 1, ..., p-1; j = 1, ..., n$)
- Criteria weights (w_j for $n = 1, ..., n$)
- Majority threshold (λ)

Number of parameters : $(2p-1)n+1$

Previous works and objective

Previous works on the elicitation of ELECTRE TRI parameters

- Several articles deals with the elicitation of parameters for classic ELECTRE TRI procedure.
  ([Mousseau and Slowinski, 1998, Mousseau et al., 2001, Ngo The and Mousseau, 2002])
- In [Leroy et al., 2011], elicitation of all the parameters of an MR-Sort model with a MIP.
- In [Cailloux et al., 2012], three MIP proposed to find a set of weights or profiles when there are multiple decision makers.

Previous works and objective

Previous works on the elicitation of ELECTRE TRI parameters

- Several articles deals with the elicitation of parameters for classic ELECTRE TRI procedure.
  ([Mousseau and Slowinski, 1998, Mousseau et al., 2001, Ngo The and Mousseau, 2002])
- In [Leroy et al., 2011], elicitation of all the parameters of an MR-Sort model with a MIP.
- In [Cailloux et al., 2012], three MIP proposed to find a set of weights or profiles when there are multiple decision makers.

Observation

- MIP require lot of time to solve simple problems
- Due to the high number of constraints and binary variables
- MIP are not suitable for problems with large amount of data

Previous works and objective

Previous works on the elicitation of ELECTRE TRI parameters

- Several articles deals with the elicitation of parameters for classic ELECTRE TRI procedure.
  ([Mousseau and Slowinski, 1998, Mousseau et al., 2001, Ngo The and Mousseau, 2002])
- In [Leroy et al., 2011], elicitation of all the parameters of an MR-Sort model with a MIP.
- In [Cailloux et al., 2012], three MIP proposed to find a set of weights or profiles when there are multiple decision makers.

Observation

- MIP require lot of time to solve simple problems
- Due to the high number of constraints and binary variables
- MIP are not suitable for problems with large amount of data

Objective of our work

- Conceive a metaheuristic allowing to learn the parameters of an MR-Sort from a large amount of data

Principe of the metaheuristic

Inputs

- n criteria ;
- p categories ;
- m assignment examples.

Output

- n criteria weights (w_j for $j = 1, ..., n$)
- $(p-1) \cdot n$ profiles evaluations ($b_{h,j}$ for $h = 1, ...p-1$; $j = 1, ..., n$)
- Majority threshold (λ)

Objective

- Maximize the classification accuracy ($CA = \frac{\text{Number of examples correctly restored}}{\text{Total number of examples}}$)
- Number of parameters to learn : $p \cdot n + 1$

Main steps of the algorithm

1. Generate $p-1$ random profiles ;
2. Learn weights and credibility threshold with a linear program
3. Improve the profiles with a metaheuristic

Inferring the weights and the majority threshold

Inputs

- $(p-1) \cdot n$ profiles evaluations ($b_{h,j}$ for $h = 1, ...p-1$; $j = 1, ..., n$)
- m assignment examples

Outputs

- n criteria weights (w_j for $j = 1, ..., n$)
- Majority threshold (λ)

Objective

- Maximizing the CA of the learning set
- Number of parameters to learn : $n + 1$

Formulation

- Problem formulated as a linear program

Inferring the weights and the majority threshold

Formulation

$$\text{Objective : min} \sum_{a_i \in A} (x_i' + y_i') \tag{1}$$

$$\sum_{\forall j | a_i S_j b_{h-1}} w_j - x_i + x_i' = \lambda \qquad \forall a_i \in A_h, h = \{2, \ldots, p-1\} \tag{2}$$

$$\sum_{\forall j | a_i S_j b_h} w_j + y_i - y_i' = \lambda - \delta \qquad \forall a_i \in A_h, h = \{1, \ldots, p-2\} \tag{3}$$

$$\sum_{j=1}^{n} w_j = 1 \tag{4}$$

$$\lambda \in [0.5; 1] \tag{5}$$

$$w_j \in [0; 1] \qquad \forall j \in F \tag{6}$$

$$x_i, y_i, x_i', y_i' \in \mathbb{R}_0^+ \qquad \forall a_i \tag{7}$$

- Max. $n + 1 + 8m$ variables (with more than 2 categories)
- Max. $m \cdot 2(p-2) + 1$ constraints (with more than 2 categories)

Inferring the profiles

Inputs

- n criteria weights (w_j for $j = \{1, ..., n\}$)
- Majority thresholds (λ)
- m assignment examples

Outputs

- $(p-1)n$ profiles evaluations ($b_{h,j}$ for $h = 1, ...p-1$, $j = 1, ..., n$)

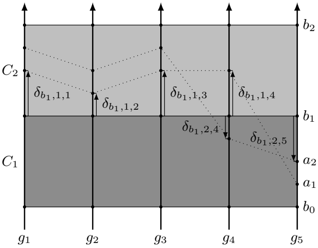
Objective

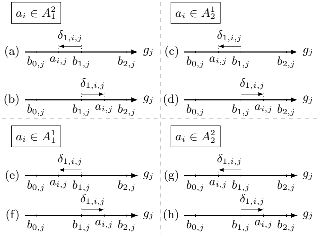
- Maximizing the CA of the learning set
- Number of parameters to learn : $(p-1)n$

Approach

- Based on a metaheuristic

Inferring the profiles - Principles





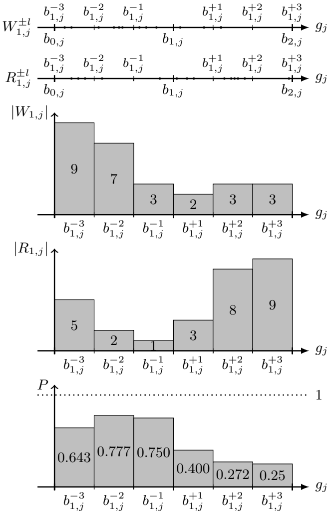
Notations

- A_1^2 (resp. A_2^1) : set of alternatives wrongly classified into C_2 (resp. C_1) instead of C_1 (resp. C_2)
- A_1^1 (resp. A_2^2) : set of alternatives correctly classified in C_1 (resp. C_2)

Example

- Fixed set of weights
- $a_1 \in A_1^2$
- $a_2 \in A_2^1$
- Profile too low and/or too high on one or several criteria

Inferring the profiles - Definition of $W_{1,j}$ and $R_{1,j}$



Notations

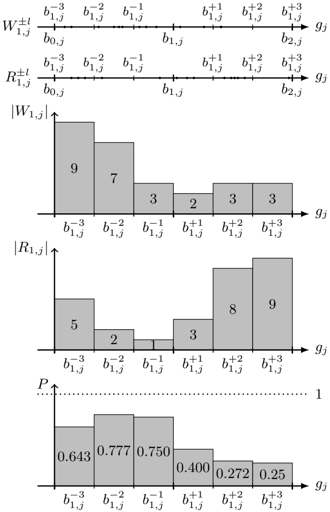
- $W_{1,j}$: Set of alternatives **wrongly** assigned by the model and for which the criterion j is **not in favor** of the correct assignment due to the profile level

$$W_{1,j} = \left\{a_i \in A_1^2 : a_{i,j} \geq b_{1,j}\right\} \cup \left\{a_i \in A_2^1 : a_{i,j} < b_{1,j}\right\} \quad (8)$$

- $R_{1,j}$: Set of alternatives **rightly** assigned by the model and for which the criterion j is **in favor** of the correct assignment due to the profile level.

$$R_{1,j} = \left\{a_i \in A_1^2 : a_{i,j} < b_{1,j}\right\} \cup \left\{a_i \in A_1^1 : a_{i,j} < b_{1,j}\right\} \cup \left\{a_i \in A_2^1 : a_{i,j} \geq b_{1,j}\right\} \cup \left\{a_i \in A_2^2 : a_{i,j} \geq b_{1,j}\right\} \quad (9)$$

Inferring the profiles - Building of the histograms



Probability function

$$P(b_{1,j}^{\pm l}) = \frac{|W_{1,j}^{\pm l}|}{|W_{1,j}^{\pm l}| + |R_{1,j}^{\pm l}|} \qquad (10)$$

Algorithm

for all $j \in \{1, ..., n\}$ **do**
  Compute $P(b_{1,j}^{\pm l})$, $\forall l$
  Find L such that
  $P(b_{1,j}^{L}) = \max_l(P(b_{1,j}^{\pm l}))$
  Draw a random number r from the uniform distribution [0, 1]
  if $r < (P(b_{1,j}^{L}))$ **then**
    $b_{1,j} = b_{1,j}^{L}$
  end if
end for

Inferring the profiles - More than two categories

Definition of $W_{h,j}$ and $R_{h,j}$

- Take into account the alternatives for which the class assigned by the DM and the model either coincide or are nearest neighbor.

$$W_{h,j} = \left\{a_i \in A_h^{h+1} : b_{h+1,j} > a_{i,j} \geq b_{h,j}\right\} \cup \left\{a_i \in A_{h+1}^{h} : b_{h-1,j} < a_{i,j} < b_{h,j}\right\} \quad (11)$$

$$R_{h,j} = \left\{a_i \in A_h^{h+1} : b_{h-1,j} \leq a_{i,j} < b_{h,j}\right\} \cup \left\{a_i \in A_h^{h} : b_{h-1,j} \leq a_{i,j} < b_{h,j}\right\} \cup \left\{a_i \in A_{h+1}^{h} : b_{h+1,j} > a_{i,j} \geq b_{h,j}\right\} \cup \left\{a_i \in A_{h+1}^{h+1} : b_{h+1,j} > a_{i,j} \geq b_{h,j}\right\} \quad (12)$$

Algorithm

- Profiles are treated in ascending order

Inferring the profiles - Parameters

Objective function and stopping criterion

- Maximization of the model's CA

Number and position of the subdivision points

- Currently, intervals between two profiles subdivided into $2k$ subintervals.
- Equal vs. unequal

Probability function

- Currently only take into account the alternatives rightly or wrongly assigned to one of the two categories neighboring the profiles

Treatment order of the profiles

- Currently treated in ascending order

Inferring all the parameters

Algorithm

- Initialize N_{mod} MR-Sort models with a set of random profiles
- Repeat at most N_o times for each of the N_{mod} models :
  1. Given the current profiles, learn the weights and a majority threshold with the linear program
  2. Given the weights and a majority threshold, improve the profiles by running the metaheuristic at most N_{it} times.
     After N_{it} loops, the profiles giving the best CA are kept.
  3. Keep the $N_{mod}/2$ best models and generate $N_{mod}/2$ new random models.
- The algorithm is stopped once a model has a CA equal to 1 or when the algorithm has run N_o times.

Objectives of the experimentations

Model retrieval

- Given a set of alternatives assigned by a known MR-Sort model
- What is the ability of the algorithm to determine the parameters of a model assigning these alternatives as much as possible to the same categories as the original model ?

Algorithm efficiency

- What is the practical complexity of the algorithm ?
- Is it able to deal with large learning sets ?
- How much time does it take to learn the parameters of a model for a given number of categories, criteria and assignment examples ?

Tolerance for error

- Learning set might contains assignment errors
- How does the algorithm react to learning sets that are not entirely compatible with a MR-Sort model ?
- Has the algorithm the ability to correct assignment errors ?

Experimental setup

Measuring algorithm efficiency

1. A random model M is generated. It includes :
   - a set of weights normalized to 1 ;
   - a set of profiles with evaluations on the n criteria between 0 and 1.
   - a majority threshold λ drawn in the interval $[0.5, 1]$.

   The assignment rule is denoted by s_M
2. A set of m alternatives with random performances on the n criteria is generated. This set is denoted by A The alternatives are assigned using the rule s_M
3. The algorithm runs and tries to maximize the number of assignments compatible with the output from step 2. The resulting model is denoted by M' and the assignment rule s'_M.
4. The value of $CA(s_M, s'_M) = \frac{|\{a \in A : s_M(a) = s_{M'}(a)\}|}{|A|}$ is computed.

Experimental setup

Measuring model retrieval

5. A set of 10000 random alternatives with random performances is generated. This set is denoted by B.
6. The alternatives contained in B are assigned using s_M and s'_M and CA of M' is computed.

Measuring tolerance for error

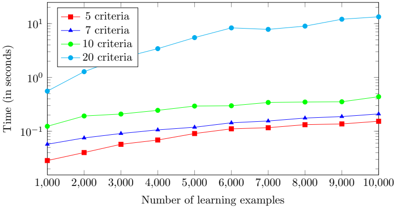
2'. A proportion of error is added in the assignment resulting from rule s_M. The rule producing the assignment with errors is denoted by $\tilde{s}_M$.

Test instances

- Each test is repeated on 10 different random instance.
- Models of 3 categories and 10 criteria.
- The values plotted are the average, the min and the max in the following graphs.

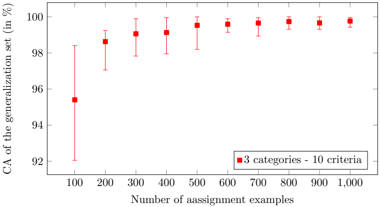
Inference of the weights and credibility threshold

Computing Time



- Increases with the number of examples
- Become important when the number of criteria increases

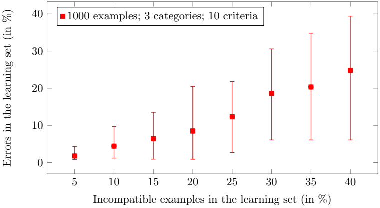
Model retrieval



- With 600 examples, $CA = 99\%$
- Good ability to restore assignments in generalization.

Inference of the weights and credibility threshold

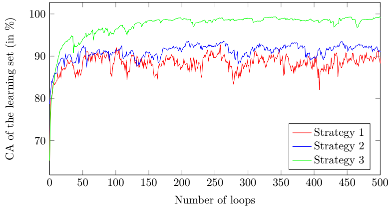
Tolerance for error



- Classification errors introduced in the learning set are corrected.
- Algorithm able to reduce the number of errors but create other errors.

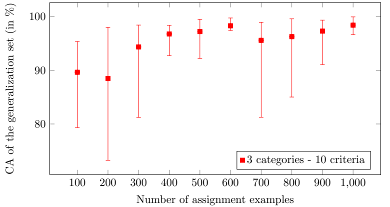
Inference of the profiles

Algorithm efficiency



1. Equally spaced subdivisions between the profiles.
2. Spacing between the subdivisions increasing as a function of the distance to the profile.
3. Spacing between subdivisions increasing as a function of the distance to the profile ; Number of intervals increasing as a function of the *CA* of the model.

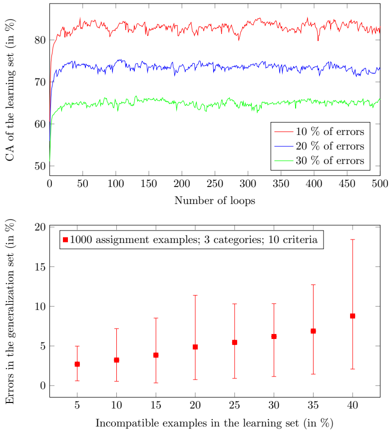
Model retrieval



- With 1000 alternatives, *CA* close to 100 %.
- Sometimes algorithm remains stuck in a local minimum.

Inference of the profiles

Tolerance for error

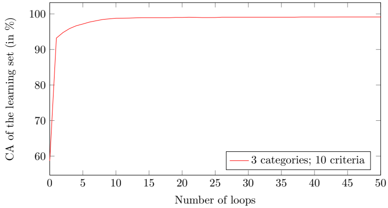


- Algorithm converges but reflects the percentage of errors.
- Algorithm identify alternatives badly assigned.

- With 10 % of errors, the algorithm restore 97% of the assignment examples.
- Generalization shows that the algorithm identify errors.

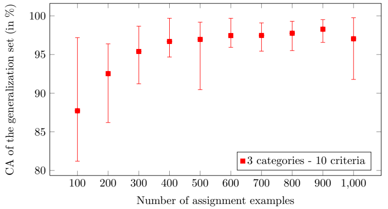
Inference of all the parameters

Algorithm efficiency



- 10 models ($N_{mod} = 10$)
- Number of loops of the metaheuristic for profiles ($N_{it} = 20$)
- After 5 loops, the algorithm has almost reached a *CA* of 100 %.

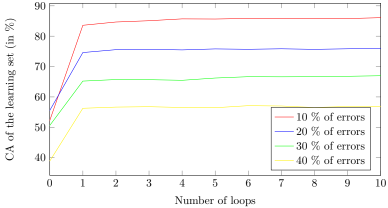
Model retrieval

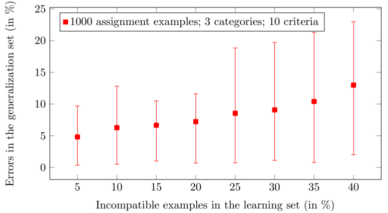


- *CA* in average bigger than 90% with more than 300 alternatives.

Inference of all the parameters

Tolerance for error





- Errors in the learning are reflected in *CA*
- Other errors are introduced

- Percentage of errors in average attenuated by the metaheuristic
- Sometimes more errors are introduced

Conclusion and further research issues

- Analysis based on models of 3 categories and 10 criteria. New experiments with more categories required.
- Need to improve the algorithm efficiency
- Use of the algorithm on a real preference learning problem
- MR-Sort model with vetoes not treated

References I

Bouyssou, D. and Marchant, T. (2007a).
An axiomatic approach to noncompensatory sorting methods in MCDM, I : The case of two categories.
European Journal of Operational Research, 178(1) :217–245.

Bouyssou, D. and Marchant, T. (2007b).
An axiomatic approach to noncompensatory sorting methods in MCDM, II : More than two categories.
European Journal of Operational Research, 178(1) :246–276.

Cailloux, O., Meyer, P., and Mousseau, V. (2012).
Eliciting ELECTRE TRI category limits for a group of decision makers.
European Journal of Operational Research, 223(1) :133–140.

References II

Leroy, A., Mousseau, V., and Pirlot, M. (2011).
Learning the parameters of a multiple criteria sorting method.
In Brafman, R., Roberts, F., and Tsoukiàs, A., editors, *Algorithmic Decision Theory*, volume 6992 of *Lecture Notes in Computer Science*, pages 219–233. Springer Berlin / Heidelberg.

Mousseau, V., Figueira, J., and Naux, J.-P. (2001).
Using assignment examples to infer weights for ELECTRE TRI method : Some experimental results.
European Journal of Operational Research, 130(1) :263–275.

Mousseau, V. and Slowinski, R. (1998).
Inferring an ELECTRE TRI model from assignment examples.
Journal of Global Optimization, 12(1) :157–174.

References III

Ngo The, A. and Mousseau, V. (2002).
Using assignment examples to infer category limits for the ELECTRE TRI method.
Journal of Multi-criteria Decision Analysis, 11(1) :29–43.

Yu, W. (1992).
Aide multicritère à la décision dans le cadre de la problématique du tri : méthodes et applications.
PhD thesis, LAMSADE, Université Paris Dauphine, Paris.