Learning the Parameters of a Multiple Criteria Sorting Method Based on a Majority Rule

Agnès Leroy[1] and Vincent Mousseau[2] and Marc Pirlot[1]

[1]MATHRO, Faculté Polytechnique, Université de Mons
9, Rue de Houdain, Mons, Belgium
marc.pirlot@umons.ac.be

[2]Laboratoire Génie Industriel, Ecole Centrale Paris
Grande Voie des Vignes 92295 Châtenay Malabry, France
vincent.mousseau@ecp.fr

Abstract. Multicriteria sorting methods aim at assigning alternatives to one of the predefined ordered categories. We consider a sorting method in which categories are defined by profiles separating consecutive categories. An alternative a is assigned to the lowest category for which a is at least as good as the lower profile of this category, for a majority of weighted criteria. This method, that we call MR-Sort, corresponds to a simplified version of ELECTRE Tri. To elicit the values for the profiles and weights, we consider a learning procedure. This procedure relies on a set of known assignment examples to find parameters compatible with these assignments. This is done using mathematical programming techniques.
The focus of this study is experimental. In order to test the mathematical formulation and the parameters learning method, we generate random samples of simulated alternatives. We perform experiments in view of answering the following questions: (a) assuming the learning set is generated using a MR-Sort model, is the learning method able to restore the original sorting model? (b) is the learning method able to do so even when the learning set contains errors? (c) is MR-Sort model able to represent a learning set generated with another sorting method, i.e. can the models be discriminated on an empirical basis?

Keywords: Multicriteria Decision Aiding, Sorting, Preference Elicitation, Learning Methods

1 Introduction

In this paper we deal with multiple criteria sorting methods that assign each alternative to a category selected in a set of ordered categories. We consider assignment rules of the following type. Each category is associated with a "lower profile" and an alternative is assigned one of the categories above this profile as soon as the alternative is at least as good as the profile for a (weighted) majority of criteria.

Such a procedure is a simplified version of ELECTRE Tri, an outranking sorting procedure in which the assignment of an alternative is determined using a more complex concordance non-discordance rule [16]. Several papers have recently been devoted to the elicitation by learning of the parameters of the ELECTRE Tri method. These learning procedures usually rely on a set of known assignment examples and use mathematical programming techniques to find parameters compatible with these assignments (see e.g. [13], [11], [14], [6]). Unfortunately, the number of parameters involved is rather high and the mathematical formulation of the constraints resulting from the assignment examples are nonlinear so that the proposed methods do not try in general to determine all parameters at the same time. They generally assume that some of these parameters are known and determine the remaining ones accordingly.

To better tackle these difficulties, we have decided to work with a simplified version of ELECTRE Tri, essentially that characterized by [1, 2]. In this version, an alternative is assigned above a limit profile if this alternative is at least as good as the profile for a sufficient coalition of criteria. We assume in addition that additive weights can be assigned to all criteria in such a way that a coalition is sufficient if the sum of the associated weights passes some majority threshold. In such a method, the parameters to be determined are the limit profiles of the categories, the criteria weights and the majority threshold.

The set of constraints on the parameters expressing the assignment of the examples, as well as other constraints, form a nonlinear mixed integer program that can be solved using CPLEX for realistic problems. Learning sets composed of up to 100 assignment examples and involving up to 5 criteria and 3 categories have been solved to optimality in a few seconds.

The interest of this study is experimental. In order to test the mathematical formulation and the parameters learning method, we have generated random samples of simulated alternatives represented by normalized performance vectors (values uniformly drawn from the [0,1] interval). We have then performed series of experiments in view of answering the following questions:

- Q1 **Model retrieval**: assuming that the examples have been assigned by means of a simulated sorting procedure based on a majority rule, does the learning method allow to elicit values of the parameters that are close to those of the original procedure used for their assignment? What size of a learning set is needed in order to obtain a "good approximation'" of these parameters?
- Q2 **Tolerance for error**: assuming that the examples have only been "approximately" assigned using a simulated sorting model, i.e. that a certain proportion of assignment errors (5 to 15%) have been introduced, to what extent do these errors perturb the elicitation of the assignment model?
- Q3 **Idiosyncrasy**: we generate an assignment model that is not based on a majority rule but on an additive value function. We assign the alternatives in the learning set according with the latter rule. The question we try to answer is whether the change in the model can be easily detected by the elicitation procedure. In other words, can the models be discriminated on an empirical basis, i.e., on the sole evidence of assignment examples?

We present the results of our experiments as well as the conclusions that we draw from them (for more detail the interested reader can refer to [10]). Further research perspectives are outlined.

2 MR-Sort: a sorting method based on a majority rule

As announced in the introduction, we depart from the usual Electre Tri sorting model that appears too complex (too many parameters) for our purpose of experimenting with a learning method. In addition, the precise procedure used for assigning alternatives to categories has not been characterized in an axiomatic manner. These are the reasons why we have turned to a simpler version of Electre Tri that has been characterized by [1, 2].

At this stage, let us assume that an alternative is just a n-tuple of elements which represent its evaluations on a set of n criteria. We denote the set of criteria by $N = \{1, \ldots, n\}$ and assume that the values of criterion i range in the set X_i. Hence the set of alternatives can be identified with the Cartesian product $X = \Pi_{i=1}^n X_i$.

According to Bouyssou and Marchant, a *non-compensatory sorting method* (NCSM) is a procedure for assigning any alternative $x \in X$ to a particular category, in a given ordered set of categories. For simplicity, assume that there are only two categories. They thus form an ordered bipartition (X^1, X^2) of X, X^1 (resp. X^2) being interpreted as the set of "bad" (resp. "good") alternatives. A sorting method (in two categories) is *non-compensatory*, in Bouyssou-Marchant sense, if the following conditions hold:

- for each criterion i, there is a partition (X_i^1, X_i^2) of X_i; X_i^1 (resp. X_i^2) is interpreted as the set of "bad" (resp. "good") levels in the range of criterion i;
- there is a family $\mathcal{F}$ of "sufficient" coalitions of criteria (i.e. subsets of N), with the property that a coalition that contains a sufficient coalition is itself sufficient;
- the set of "good" levels $\mathcal{X}_i^2$ on each criterion and the set of sufficient coalitions $\mathcal{F}$ are such that alternative $x \in X$ belongs to the set of "good" alternatives X^2 iff the set of criteria on which the evaluation of x belongs to the set of "good" levels is a sufficient coalition, i.e.:

$$x = (x_1, \dots, x_i, \dots, x_n) \in X^2 \quad \text{iff} \quad \{i \in N | x_i \in X_i^2\} \in \mathcal{F}. \tag{1}$$

Non compensatory sorting models have been fully characterized by a set of axioms in the case of two categories [1]. [2] extends the above definition and characterization to the case of more than two categories. These two papers also contain definitions and characterizations of NCSM with vetoes.

In the present paper we consider a special case of the NCSM model (with two or more categories and no veto). The Bouyssou-Marchant models are specialized in the following way:

1. We assume that X_i is a subset of $\mathbb{R}$ (e.g. an interval) for all $i \in N$ and the partitions (X_i^1, X_i^2) of X_i are compatible with the order on the real numbers $<$, i.e., for all $x_i \in X_i^1$, $x_i' \in X_i^2$, we have $x_i < x_i'$. We assume furthermore that X_i^2 has a smallest element b_i, which implies that $x_i < b_i \leq x_i'$.
2. There is a weight w_i associated with each criterion and a threshold λ such that a coalition is sufficient iff the sum of the weights of the criteria belonging to the coalition passes threshold λ: for all subset F of N, $F \in \mathcal{F}$ iff $\sum_{i \in N} w_i \geq \lambda$; we may assume w.l.o.g. that the weights are normalized ($\sum_{i \in N} w_i = 1$).

Rule (1) can thus be rephrased as:

$$x = (x_1, \dots, x_i, \dots, x_n) \in X^2 \quad \text{iff} \quad \sum_{i \in N : x_i \geq b_i} w_i \geq \lambda. \tag{2}$$

To bridge the gap with the classical Electre Tri model, let us consider that A is the set of alternatives and $g_i : A \to \mathbb{R}$ are functions associating each alternative $a \in A$ its evaluation on criterion i. Alternative a is hence represented by the n-tuple $(g_1(a), \dots, g_i(a), \dots, g_n(a)) \in X = \Pi_{i=1}^n X_i$. A is partitioned into two categories (A^1, A^2), with A^1 (resp. A^2) the set of "bad" (resp. "good") alternatives. We extend rule (2) to sets of alternatives having vectors in X as their evaluation on the n criteria and we assume that (A^1, A^2) satisfies the extension of rule (2), namely:

$$a \in A^2 \quad \text{iff} \quad \sum_{i \in N : g_i(a) \geq b_i} w_i \geq \lambda. \tag{3}$$

Clearly, (3) is also a particular case of the classical ELECTRE TRI (pessimistic) assignment rule.

In rules (2) or (3), the b_i's compose a vector $b \in \mathbb{R}^n$, which is the (lower) limit profile of category A^2. An alternative a belongs to A^2 iff its evaluations $g(a_i)$ are at least as good as b_i on a subset of criteria that has sufficient weight.

In the sequel, we call a model that assigns alternatives to (two) categories according to rule (3) a *Majority Rule Sorting Model* (MR-Sort). The parameters of such a model are the n components of the limit profile b, the weights of the criteria $w_1, \ldots w_n$ and the majority threshold λ, in all $2n+1$ parameters.

This setting can easily be generalized to sorting in k categories $(A^1, \ldots, A^h, \ldots, A^k)$ forming an ordered partition of A. The MR-Sort assignment rule is the following. Alternative $a \in A$ is assigned to category A^h, for $h = 2, \ldots k-1$ if

$$\sum_{i \in N: g_i(a) \geq b_i^{h-1}} w_i \geq \lambda \quad \text{and} \quad \sum_{i \in N: g_i(a) \geq b_i^h} w_i < \lambda \tag{4}$$

where $b^h = (b_1^h, \ldots b_n^h)$ is the lower limit profile of category A^h. Alternative a is assigned to category A^1 if $\sum_{i \in N: g_i(a) \geq b_i^1} w_i < \lambda$. It is assigned to category A^k if $\sum_{i \in N: g_i(a) \geq b_i^{k-1}} w_i \geq \lambda$. A MR-Sort model with k categories involves $kn+1$ parameters ($k-1$ limit profiles, the weight vector and the majority threshold).

3 Learning a MR-Sort model

Assuming that the decision maker is able to provide us with a number of a priori assignments of alternatives to categories, we may take advantage of this information to restrict the set of MR-Sort models compatible with such an information and possibly select one or several typical ones among them. Let $A^* \subseteq A$ be the subset of alternatives assigned to categories by the decision maker. A^* will be referred to as the *learning set*. Let $A^* = \{a_1, \ldots, a_j, \ldots, a_{na}\}$, where na is the number of alternatives in the learning set. In the case of two categories the DM's assignments result in a bipartition (A^{*1}, A^{*2}) of the learning set into a set of "bad" and "good" alternatives, respectively. These assignments generate constraints on the parameters of the MR-Sort models. Below, these constraints receive a linear formulation and are integrated into a mixed integer linear program (MIP) that is designed to select a particular feasible set of parameters.

3.1 The case of two categories

We consider the case involving two categories separated by a frontier denoted b (for more than two categories see section 3.2). For $a_j \in A^*$, let us define the binary variables δ_{ij} $(i = 1, \ldots, n)$ such that $\delta_{ij} = 1 \Leftrightarrow g_i(a_j) \geq b_i$ and $\delta_{ij} = 0 \Leftrightarrow g_i(a_j) < b_i$. For δ_{ij} to be defined consistently, we impose the following constraints (M being an arbitrary large positive value):

$$\begin{cases} M(\delta_{ij} - 1) \leq \; g_i(a_j) - b_i \; < M \cdot \delta_{ij} \\ \qquad\qquad\quad \delta_{ij} \in \{0, 1\} \end{cases} \tag{5}$$

Using the δ_{ij} binary variables, we define continuous variables c_{ij} such that $c_{ij} = 0 \Leftrightarrow \delta_{ij} = 0$ and $c_{ij} = w_i \Leftrightarrow \delta_{ij} = 1$ (where w_i denotes the weight of criterion g_i). To do so, we impose that $\delta_{ij} - 1 + w_i \leq c_{ij} \leq \delta_{ij}$ and $0 \leq c_{ij} \leq w_i$.

As we consider two categories, the set of assignment examples is defined by two subsets $A^{*1} \subset A$ and $A^{*2} \subset A$; A^{*1} (A^{*2}, respectively) is composed of alternatives which the DM intuitively assigns to the good (resp. bad) category. In order for these assignment examples to be reproduced by the MR-sort model, the constraints (6) should be posed.

$$\begin{cases} \sum_{i \in N} c_{ij} < \lambda, \forall a_j \in A^{*1} \\ \sum_{i \in N} c_{ij} \geq \lambda, \forall a_j \in A^{*2} \end{cases} \tag{6}$$

In order to discriminate among the MR-sort models compatible with the preference information provided by the DM (assignment examples A^{*1} and A^{*2}), we consider the objective function which maximize the robustness of assignment examples, as defined in (6). To do so, we introduce additional continuous variables, x_j and y_j, for each $a_j \in A^*$, and α defined in (7). Hence, maximizing $z = \alpha$ amounts to maximizing the value of the minimal slack in the constraints (6). Strict inequalities are transformed into non-strict ones by introducing an arbitrary small positive quantity ε.

$$\begin{cases} \sum_{i \in N} c_{ij} + x_j + \varepsilon = \lambda & \forall a_j \in A^{*1} \\ \sum_{i \in N} c_{ij} = y_j + \lambda & \forall a_j \in A^{*2} \\ \alpha \leq x_j & \forall a_j \in A^* \\ \alpha \leq y_j & \forall a_j \in A^* \end{cases} \tag{7}$$

This leads us to the following mathematical program :

$$\begin{cases} max\ \alpha & \\ \quad \sum_{i \in N} c_{ij} + x_j + \varepsilon = \lambda & \forall a_j \in A^{*1} \\ \quad \sum_{i \in N} c_{ij} = \lambda + y_j & \forall a_j \in A^{*2} \\ \quad \alpha \leq x_j,\ \alpha \leq y_j & \forall a_j \in A^* \\ \quad c_{ij} \leq w_i & \forall a_j \in A^*, \forall i \in N \\ \quad c_{ij} \leq \delta_{ij} & \forall a_j \in A^*, \forall i \in N \\ \quad c_{ij} \geq \delta_{ij} - 1 + w_i & \forall a_j \in A^*, \forall i \in N \\ \quad M\delta_{ij} + \varepsilon \geq g_i(a_j) - b_i & \forall a_j \in A^*,\ \forall i \in N \\ \quad M(\delta_{ij} - 1) \leq g_i(a_j) - b_i & \forall a_j \in A^*,\ \forall i \in N \\ \quad \sum_{i \in N} w_i = 1,\ \lambda \in [0.5, 1] & \\ \quad w_i \in [0, 1] & \forall i \in N \\ \quad c_{ij} \in [0, 1],\ \delta_{ij} \in \{0, 1\} & \forall a_j \in A^*,\ \forall i \in N \\ \quad x_j, y_j \in \mathbb{R} & \forall a_j \in A^* \\ \quad \alpha \in \mathbb{R} & \end{cases} \tag{8}$$

3.2 More than 2 categories

It is not difficult to modify program (8) in order to deal with more than two categories. We consider the general case in which k categories are defined by $k-1$ limit profiles b^1, b^2, ..., b^h, ..., b^{k-1} (where $b^h = (b^h_1, ...b^h_n)$). For each alternative a_j in category A^{*h} of the learning set A^* (for $h = 2, \ldots, k-1$), we introduce $2n$ binary variables δ^h_{ij} and δ^{h-1}_{ij}, for $i = 1, \ldots, n$. We force δ^l_{ij} to be equal to 1 iff $g_i(a_j) \geq b^l_i$ for $l = h-1, h$ and $\delta^h_{ij} = 0 \Leftrightarrow g_i(a_j) < b^h_i$. We introduce $2n$ continuous variables c^l_{ij} ($l = h-1, h$) constrained to be equal to w_i if $\delta^l_{ij} = 1$ and to 0 otherwise (as is done in (8)). Finally, we express that a_j is at least as good as profile b_{h-1} on a subset of criteria that has sufficient weight while this is not true w.r.t. profile b_h; we write constraints similar to (6) to express this.

The case in which a_j belongs to one of the extreme categories (A^{*1} and A^{*k}) is simpler. It requires the introduction of only n binary variables and n continuous variables. Indeed if a_j belongs to A_1 we just have to express that the subset of criteria on which a_j is at least as good as b_1 has not sufficient weight. In a dual way, when a lies in A^k, the best category, we have to express that it is at least as good as the upper profile b^k on a subset of criteria that has sufficient weight.

3.3 Infeasible learning sets

The MIP programs presented in the two previous subsections may prove infeasible in case the assignments of the alternatives in the learning set are incompatible with all MR-sort models. In order to be able to tackle such problems we formulate a MIP that finds a MR-sort model maximizing the number of alternatives in the learning set that the model correctly assigns.

In the two categories case, for each $a_j \in A^*$, we introduce a binary variable γ_j which is equal to one if alternative a_j is correctly assigned by the MR-Sort model, and equal to zero otherwise. To ensure that the γ_j variables are correctly defined, we modify the constraints (6) in the following way:

$$\begin{cases} \sum_{i \in N} c_{ij} < \lambda + M(1-\gamma_j),\ \forall a_j \in A^{*1} \\ \sum_{i \in N} c_{ij} \geq \lambda - M(1-\gamma_j),\ \forall a_j \in A^{*2} \end{cases} \tag{9}$$

Starting from (8) and substituting constraints (6) by (9), and the objective function by the new objective $z = \sum_{a_j \in A^*} \gamma_j$, we obtain a MIP that yields a subset $A^{*\prime} \subseteq A^*$ of maximal cardinality that can be represented by an MR-Sort model. A generalization to more than two categories is obtained by bringing similar changes to the model described in section 3.2. These models will be used in the second and third experiments below (sections 4.1 and 4.2).

4 Empirical design and results

Our goal is to test the "learnability" of the MR-Sort model based on the previous MIP formulation. The three issues raised in the introduction, namely, model retrieval, tolerance for error, and idiosyncrasy are investigated through simulations. Such simulations involve generating alternatives, simulating a DM assigning these alternative, and learning an MR-Sort model from this information.

4.1 Experiment 1: model retrieval

Our strategy is the following. We generate a set of alternative, and a hypothetical MR-sort model, denoted M. Then we simulate the behavior of a DM assigning the generated alternatives, while having this MR-Sort model in mind. Hence, we constitute a learning set A^* by assigning the generated alternatives using the MR-Sort model. We infer an MR-Sort model M' compatible with the learning set using the MIP formulation presented in section 3. Although these models may be quite different, they coincide on the way they assign elements of A^*, by construction. In order to compare models M and M', we randomly generate a large set of other alternatives and we compute the percentage of "assignment errors", i.e. the proportion of these alternatives that models M and M' assign to different categories.

For small learning sets, it is expected that the inferred model is rather arbitrary. Therefore, our first experiment aims at investigating the following two issues:

– What is the typical size of a learning set which would lead to an inferred model "close" to the original one, i.e. yielding a small percentage of assignment errors?
– Does the MIP inference program remain tractable when the size of the problem increases ?

Generating alternatives A set of `na` alternatives is generated. Each alternative is identified with a vector drawn from the unit hypercube $[0,1]^n$ (uniform distribution). Such a vector represents the evaluations of an alternative along n criteria. Note that these evaluations are independent random variables. Drawing the evaluations from the [0,1] interval is not restrictive, since the assignment rule is invariant up to a strictly increasing transformation of the scale of each criterion.

Simulating an MR-Sort model n numbers are randomly drawn from the unit interval, then normalized, yielding weights w_i associated to each criterion i. A majority threshold λ is is randomly drawn from the $[0.5, 1]$ interval. To generate the $k-1$ vector profiles $b_h = (b_1^h, , \ldots, b_i^h, \ldots, b_n^h)$, we proceed as follows. For $i = 1, ..., n$, b_i^1 is randomly drawn from the $[0, \frac{2}{k}]$ interval, b_i^2 is randomly drawn from the $[b_i^1, \frac{3}{k}]$ interval; generally, b_i^h is randomly drawn from the $[b_i^{h-1}, \frac{h+1}{k}]$ interval, for $h = 2, ..., k-1$. In this way, we guarantee that each vector profile b^h dominates the previous one b^{h-1}. Moreover, for $h = 1, ..., k-1$, b_i^h divide the $[0,1]$ interval scale of criterion i in sub-interval that are of similar length (roughly speaking $\frac{1}{k}$ on average).

Empirical design We run 10 instances of each of the problems obtained by varying the following parameters:

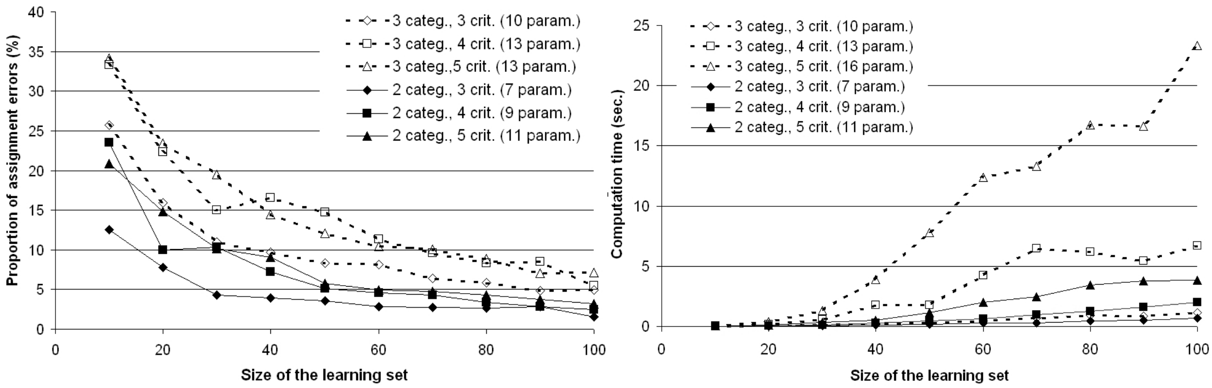
- Two, three categories,
- Three, four, five criteria,
- Learning sets containing 10 to 100 alternatives.

We use CPLEX to solve the MIP model (for several categories) and infer an MR-Sort model. We record the CPU time used. We also compute the proportion of vectors from a set B of 10 000 randomly generated alternatives that are assigned to the same category by the initial and inferred models. The lower this proportion, the greater the discrepancy between the original and inferred models.

Results The main results of these experiments are summarized in figure 1. Figure 1a shows the average percentage of assignment errors as a function of the size of the learning set (from 10 to 100). The percentage of assignment errors is the percentage of the 10 000 alternatives in B that are not assigned to the same category by the original and the inferred model. The percentage represented is the *average* of the percentages observed on 10 learning sets instances. Figure 1b represents the computing time used to learn the inferred model as a function of the size of the learning set (all computations have been performed on a single standard PC).

Comments On figure 1a we see that the assignment errors tend to decrease when the size of the learning set increases, in each simulation setting. This obviously means that the higher the cardinality of the learning set, the more determined the model. For a given size of the learning set, the error rate grows with the number of model parameters (number of criteria, number of categories). To guarantee an error rate errors inferior to 10%, we typically need learning sets consisting of 40 (resp. 70) alternatives in the case of 2 (resp. 3) categories, 5 criteria). In other words, small learning sets (e.g. 10 or 20), do not allow to retrieve the original model parameters with good precision. Experiments of this type allow to estimate the size of the learning sets that yield a given level of the error rate.

Computing times (figure 1b) remain under 10 seconds on average for 2 categories and for 3 categories up to 4 criteria, even for the largest learning sets we consider (100 alternatives). However,



(a) Assignment errors (%) (b) CPU time

Fig. 1: Results of experiment 1

the case of 3 categories and 5 criteria suggests that computing time could soon become unacceptable when the number of categories and/or criteria takes larger values (about 25 seconds for 100 alternatives in the learning set in the case of 3 categories and 5 criteria).

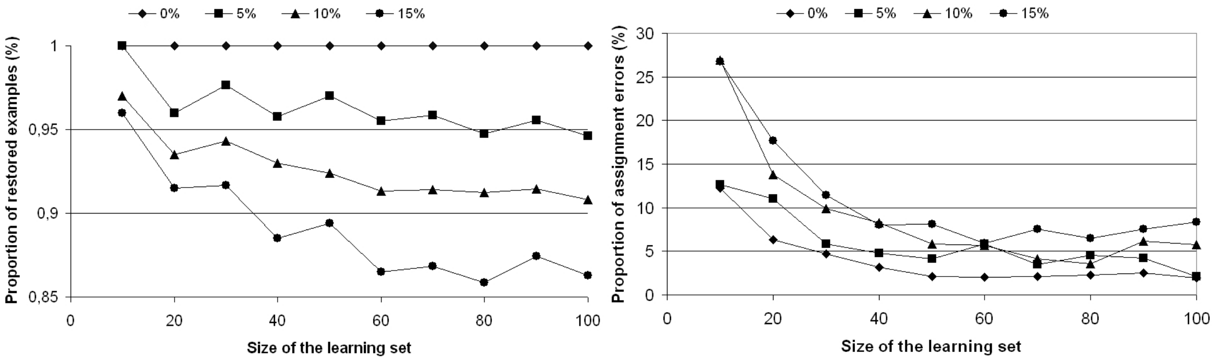
4.2 Experiment 2: tolerance for error

In this second experiment we study to which extent the model is "learnable" when the assignment examples are "approximately" simulated by an MR-Sort model, i.e. that a proportion of assignment errors has been introduced in the learning set. To do so, we randomly generate sets of `na` alternatives and an MR-Sort model as described in section 4.1. We then modify the obtained learning set by introducing a proportion of 5%, 10% and 15% of errors. These errors correspond to alternatives in the learning set which are randomly assigned to a category which differs from the one computed by the MR-Sort model.

As these learning sets include a proportion of erroneous assignments, it is possible that no MR-Sort model can restore the whole learning set. Therefore we compute the maximum number of alternatives of the learning set whose assignments is compatible with an MR-Sort model. This is done using the MIP formulation given in section 3.3. As in the first experiment, we run 10 instances of each of the problems obtained by varying the size of the learning sets (from 10 to 100 alternatives), with a fixed number of categories (two) and criteria (three).

Figure 2a represents the ratio of the maximal value of the objective function of the MIP by the size of the learning set, i.e. the maximal proportion of alternatives in the learning set whose assignments are compatible with a MR-Sort model (as a function of the percentage of assignment errors made in assigning the alternatives in the learning set to a category). Figure 2b represents the proportion of randomly generated evaluation vectors that are assigned to the same category by the initial and the inferred MR-Sort model. Figure 3 shows how the CPU time evolves with the size of the learning set and the proportion of errors.

Comments On figure 2a we observe that the the maximal proportion of alternatives in the learning set whose assignments are compatible with a MR-Sort model decrease from a high value to reach



(a) Optimal objective value (%) (b) Assignment errors (%)

Fig. 2: “Learnability” of an MR-Sort model (2 categories, 3 criteria) using learning sets involving errors

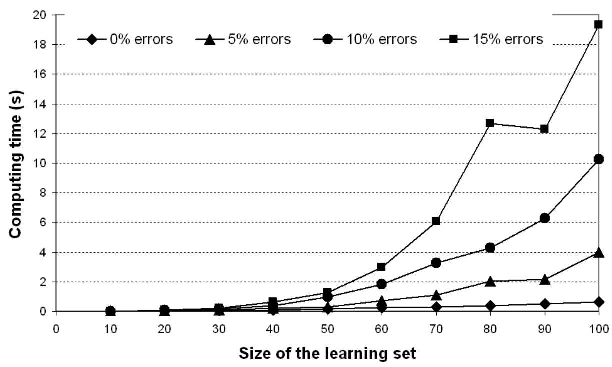


Fig. 3: Computing time in Experiment 2

asymptotically a minimum, when the size of the learning set increases. Moreover, it should be noted that, when the learning set is large, the proportion of restored examples in learning sets containing 5% (10%, 15%, respectively) errors approximately corresponds to 95% (90%, 85%, respectively). This means that, when the learning set is small, the MR-Sort model is flexible enough to reproduce almost all the learning set despite the errors; however, when the size of the learning set is large, as the MR-Sort model is more specific, the proportion of alternatives in the learning set whose assignment is not reproduced by the inferred model corresponds to the proportion of errors introduced in the learning set. Note however that alternatives in the learning set that are excluded when inferring the model do not necessarily correspond to the errors introduced in the learning set. However, the proportion of alternatives excluded when inferring the model is at most equal to the proportion of introduced errors.

On figure 2b we see that the proportion of randomly generated evaluation vectors that are assigned to different categories by the initial and the inferred MR-Sort model decreases with the size of the learning set, independently of the proportion of errors in the learning set. For sufficiently large learning sets (40 alternatives or more), the presence of errors in the learning set deteriorates the ability of the model to restore the assignment of random alternatives, but in a limited way only. For instance, a model inferred using a learning set of 100 alternatives with 15% errors induces 8% incorrect assignments, while a model inferred using a learning set of the same size with no error induce 2% incorrect assignments. It appears that the presence of a limited number of errors in the learning set does not strongly impact the "learnability" of the model.

Figure 3 shows that the CPU time increases with the size of the learning set, for all proportion of errors in the learning set. Moreover, for large learning sets (more than 50 alternatives) the proportion of errors in the learning set significantly impacts the CPU time. Although this experiment considers datasets with two categories and four criteria only, the average computing time with a learning set of 100 alternatives and 15% errors is approximately 20 seconds. Moreover, it should be recalled that the CPU time also increases in the case of error-free learning sets with the number of categories and criteria. This suggests that the inference program using a learning set with error might become intractable when the number of criteria and categories increases.

4.3 Experiment 3: idiosyncratic behavior

In the third experiment, we have tried to see to what extent an MR-Sort model is able to account for assignments made by another–definitely different–sorting model. In view of this, we have generated a sorting model based on an additive value function (AVF-Sort model). Such a model is used e.g. in the UTADIS method [5, 17]. We generate such a model by slightly modifying the procedure designed for generating a MR-Sort model (see section 4.1).

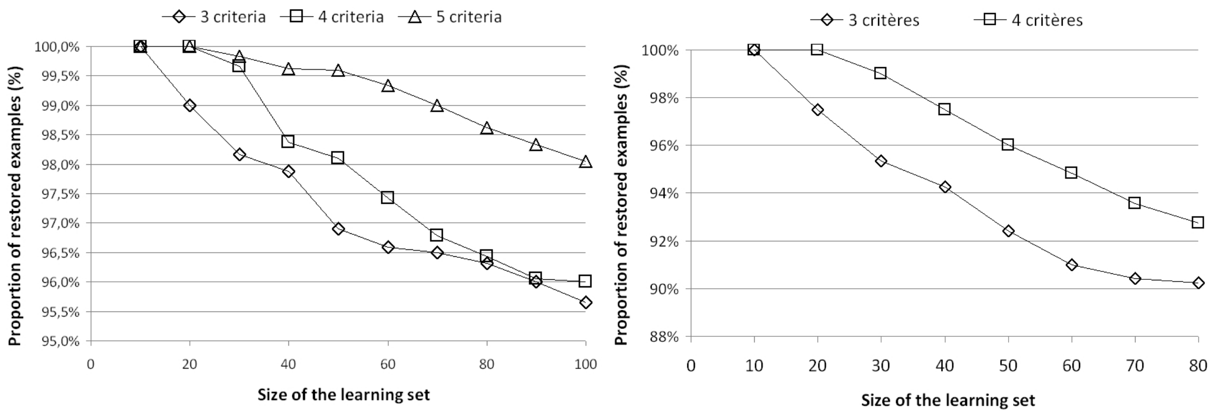
Simulating an AVF-Sort model We generate weights and profiles as for the MR-Sort model. For each vector profile $b_h = (b_1^h, \ldots, b_i^h, \ldots, b_n^h)$, we compute an associated threshold $\beta^h = \sum_{i=1}^n w_i b_i^h$. Then we assign alternatives to categories by means of the following rule. Alternative $a = (g_1(a), \ldots, g_n(a))$ is assigned to category A^h, for $h = 2, \ldots, k$ if

$$\beta^{h-1} \leq \sum_{i \in N} w_i g_i(a) < \beta^h; \tag{10}$$

alternative a is assigned to category A^1 if $\sum_{i \in N} w_i g_i(a) < \beta^1$; it is assigned to category A^k if $\beta^{k-1} \leq \sum_{i \in N} w_i g_i(a)$.

As in the previous experiments, we assign 10 to 100 alternatives considered as forming a learning set to categories, using an AVF-Sort model. Then we run the MIP described in section 3.3 to learn an MR-Sort model that assigns as many as possible of these alternatives to the same category as the AVF-Sort model.

Results Figure 4 shows the proportion of alternatives in the learning set that a learned MR-Sort model has been able to assign to the same category as the original AVF-Sort model. Figure 4a shows the results for two categories and 3 to 5 criteria. Figure 4b shows similar results for three categories and 3 and 4 criteria. In the latter case, the maximal size of learning set is 80 (for larger size, computing times become excessive).



(a) Two categories (b) Three categories

Fig. 4: Results of experiment 3: maximal proportion of alternatives in learning set compatible with a MR-Sort model

Comments It may come as a surprise that MR-Sort models are flexible enough to accommodate more than 95% (resp. 90%) of the alternatives assigned by the AVF-Sort model when there are two (resp. three) categories. Hence, it seems uneasy to detect, on the sole basis of the assignment of the alternatives in a learning set, which sorting model has been used to generate the learning set.

Another observation is the following. The larger the number of criteria, the higher the proportion of alternatives in the learning set that can be assigned consistently by the two models. This is surely due to the higher number of degrees of freedom (parameters) in the models when there are more criteria.

No extensive experimentation has been performed sofar on the way the learned MR-Sort model behaves when its assignments are compared to those of the original AVF-model on a large sample of generated alternatives. This has only be checked in the case of two categories and three criteria on a set of $100,000$ generated alternatives. The proportion of these alternatives assigned to different classes by the two models amounts to 15.4%, a proportion significantly more important than that observed on the learning set.

5 Conclusion

This paper has experimentally investigated the feasibility of eliciting the parameters of an MR-Sort model based on a set of assignment examples. It has explored how the mathematical programs involved in the elicitation respond when exposed to three different learning settings.

The main insight resulting from these experiments is that eliciting an MR-Sort model is highly demanding in terms of information. A large number of assignment examples are required to reliably elicit such sorting models: from around 20 examples for the 7 parameters model with 2 categories and 3 criteria to around 75 examples for the 16 parameters model with 3 categories and 5 criteria; learning sets of that size guarantee less than 10% assignment errors when using the learned model. The need for much information is also attested by the fact that the MR-Sort model is able to accommodate both learning sets with errors and sets of assignments made using a different model.

This has implications for practice, which go beyond the learning of an MR-Sort model. We develop some of these implications below as well as some further research issues.

Parsimony in the choice of a model The scarcity of available information may (should ?) drive the analyst from considering models involving many parameters (such as the classical ELECTRE TRI model) to simpler ones (such as MR-Sort). Our experiments tend to show that the expressive power of MR-Sort is more than sufficient when the learning set contains a few dozens of assignment examples. In such circumstances, the learning of models involving more parameters is likely to yield highly arbitrary results, since the learned model will be one among many possible models which equally well reproduce the assignment examples. Even in the context of the simple MR-Sort model, it could be advisable to model the decision problem using few categories when the preference information is scarce. Indeed reducing the number of categories–provided this is an option–will strongly influences the number of parameters to be elicited. The degree of refinement of the categorizations should be related to the number of assignment examples that the DM is expected to provide.

Questioning about parameters In this work, we have only used assignment examples for learning the parameters of the model. To reduce the complexity of the learning process, one may think of directly asking the DM for the value of some of the parameters, such as the profiles or the weights, and learn the others. This approach has been proposed for ELECTRE TRI in [14, 12]. It is however exposed to criticism since one cannot assume that the DM is aware of the meaning of the technical parameters of a method[1]; the cognitive value of spontaneous answers, for instance about the weights of criteria, is questionable (see [3], section 4.4). In principle, questions about parameters, asked to the DM, should preferably be formulated in terms of comparisons of alternatives, assignment to categories, etc, i.e. in terms of objects and issues related to the real decision problem.

Working with all models compatible with the assignment examples In view of the vastly undetermined character of the parameters in preference models learned on the basis of examples (not only in ordered assignment problems but also in ranking problems), it has been advocated ([8, 9, 7]) to work with all models compatible with the available information (assignment examples in a sorting problem, pairs belonging to the preference relation in a ranking problem, restriction on the range of parameters, etc). Valid recommendations hence are basically those shared by all models compatible with the information. Our experiments challenge the operational character of such an approach, also referred to as "Robust Ordinal Regression". It is likely that, unless the available information is very rich or the domain of variation of the parameters severely restricted, the conclusions compatible with all possible models will be very poor. In any case, this approach calls for empirical validation of its operational character. Note that experimental results similar to those presented in the present paper have been obtained for ranking problems under the additive value function model (see [15]).

Introducing vetoes in the MR-Sort model As we have seen with our second experiment, considering sets of assignment examples that imperfectly follow an MR-Sort model leads to learned models that incorrectly assign some examples. In some cases, these examples have not been correctly assigned due to the fact that they are too much below the level of their category bottom profile on some criteria. Introducing vetoes in the MR-Sort model is a simple way of fixing such situations. Although

[1] For instance, weights of criteria represent tradeoffs in an additive value function model, while they are used to measure the strength of coalitions of criteria in the ELECTRE methods.

it is possible indeed to learn both the parameters of a MR-Sort model and the veto threshold at a time (a mathematical program doing that is proposed in [10]), one could think to proceeding in steps. First elicit the MR-Sort model that best fits the assignment examples. Then examine the incorrectly assigned examples and see whether such incorrect assignments may be caused by veto effects. Finally, estimate the veto thresholds. An alternative approach whose objective is to minimize the number of criteria on which veto occurs is proposed in [4]. Further work should be devoted to developing the appropriate tools for estimating veto thresholds.

Selecting informative assignment examples In our experiments, the assignment examples were generated randomly. Since the amount of information contained in the examples in view of determining the model is a crucial issue, one may want in practice to select as informative as possible assignment examples. An example is all the more informative that it strongly reduces the set of parameters compatible with its assignment. Developing a methodology for efficiently (i.e. by means of questions the answer of which is as informative as possible) eliciting sorting or ranking models by learning is an interesting research challenge.

In view of the issues raised above, we hope to have convinced the reader that the experimental analysis of learning methods in MCDA is a subject that has not received enough attention sofar. We believe that it deserves further efforts and we have tried to suggest a few new research directions.

Acknowledgment

We thank two anonymous referees for helpful comments. The usual caveat applies.

References

1. Bouyssou, D., Marchant, T.: An axiomatic approach to noncompensatory sorting methods in MCDM, I: The case of two categories. European Journal of Operational Research 178(1), 217–245 (2007)
2. Bouyssou, D., Marchant, T.: An axiomatic approach to noncompensatory sorting methods in MCDM, II: More than two categories. European Journal of Operational Research 178(1), 246–276 (2007)
3. Bouyssou, D., Marchant, T., Pirlot, M., Tsoukiàs, A., Vincke, P.: Evaluation and decision models with multiple criteria: Stepping stones for the analyst. International Series in Operations Research and Management Science, Volume 86, Boston (2006)
4. Cailloux, O., Meyer, P., Mousseau, V.: Eliciting Electre Tri category limits for a group of decision makers. Tech. rep., Laboratoire Génie Industriel, Ecole Centrale Paris (June 2011), Cahiers de recherche 2011-09
5. Devaud, J., Groussaud, G., Jacquet-Lagrèze, E.: UTADIS: Une méthode de construction de fonctions d'utilité additives rendant compte de jugements globaux. In: European working group on MCDA, Bochum, Germany (1980)
6. Dias, L., Mousseau, V.: Inferring ELECTRE's veto-related parameters from outranking examples. European Journal of Operational Research 170(1), 172–191 (2006)
7. Greco, S., Kadziński, M., Mousseau, V., Słowiński, R.: ELECTRE-GKMS: Robust ordinal regression for outranking methods. European Journal of Operational Research 214(10), 118–135 (2011)
8. Greco, S., Mousseau, V., Słowiński, R.: Ordinal regression revisited: multiple criteria ranking using a set of additive value functions. European Journal of Operational Research 191(2), 415–435 (December 2008)
9. Greco, S., Mousseau, V., Słowiński, R.: Multiple criteria sorting with a set of additive value functions. European Journal of Operational Research 207(3), 1455–1470 (2010)

10. Leroy, A.: Apprentissage des paramètres d'une méthode multicritère de tri ordonné (2010), Master Thesis, Université de Mons, Faculté Polytechnique
11. Mousseau, V., Figueira, J., Naux, J.: Using assignment examples to infer weights for ELECTRE TRI method: Some experimental results. European Journal of Operational Research 130(2), 263–275 (Apr 2001)
12. Mousseau, V., Figueira, J., Naux, J.: Using assignment examples to infer weights for ELECTRE TRI method: Some experimental results. European Journal of Operational Research 130(2), 263–275 (2001)
13. Mousseau, V., Slowinski, R.: Inferring an ELECTRE TRI model from assignment examples. Journal of Global Optimization 12(2), 157–174 (1998)
14. Ngo The, A., Mousseau, V.: Using Assignment Examples to Infer Category Limits for the ELECTRE TRI Method. Journal of Multiple Criteria Decision Analysis 11(1), 29–43 (2002)
15. Pirlot, M., Schmitz, H., Meyer, P.: An empirical comparison of the expressiveness of the additive value function and the Choquet integral models for representing rankings. In: 25th Mini-EURO Conference Uncertainty and Robustness in Planning and Decision Making (URPDM 2010) (2010)
16. Roy, B., Bouyssou, D.: Aide multicritère à la décision: méthodes et cas. Economica Paris, Paris (1993)
17. Zopounidis, C., Doumpos, M.: PREFDIS: a multicriteria decision support system for sorting decision problems. Computers & Operations Research 27(7-8), 779–797 (June 2000)