Learning an MR-sort model from data with latent criteria preference direction

Pegdwendé Minoungou, Vincent Mousseau, Wassila Ouerdane, Paolo Scotton

Learning an MR-Sort model from data with latent criteria preference directions

Pegdwendé Minoungou[1,2], **Vincent Mousseau**[1], **Wassila Ouerdane**[1], **Paolo Scotton**[3]

Abstract. The Majority Rule Sorting (MR-Sort) method assigns alternatives evaluated on multiple criteria to one of the predefined ordered categories. The Inverse MR-Sort problem (Inv-MR-Sort) consists in computing MR-Sort parameters that match a dataset. Existing learning algorithms for Inv-MR-Sort consider monotone preference on criteria. We extend this problem to the case where the preference directions on criteria are not known. We propose and test an algorithm that learns from the training data the preference direction of each criterion together with the other MR-Sort parameters.

1 Introduction

A computer-products retail company is distributing a new Windows tablet, and wants to send targeted marketing emails to clients who might be interested in this new product. To do so, clients are to be classified into two categories: *potential buyer* and *not interested*. To avoid spamming, only clients in the former category will receive an email. To sort clients, four clients characteristics are considered as criteria, all of them being homogeneous to a currency e.g. € : the turnover over the last year of *(i)* Windows PC, *(ii)* Pack Office, *(iii)* Linux PC, and *(iv)* Dual boot PC.

As the company wants to advertise a new Windows tablet, both first two criteria are to be maximized (the more a client buys Windows PCs and Pack Office, the more he/she is interested in products with a Windows system), and the third criterion is to be minimized (the more a client buys Linux PCs, the less he/she is interested in products with a Windows system). The marketing manager is convinced that the last criterion should be taken into account, but does not know whether it should be maximized or minimized; a subset of clients has been partitioned into not interested/potential buyer. Based on this dataset, the goal is to simultaneously learn the classifier parameters and the preference direction for the last criterion.

As illustrated in the example above, this paper considers multicriteria sorting problems in which alternatives are to be assigned to one of the p predefined ordered categories $C^1, \ldots, C^p$. Among the existing multicriteria sorting methods [6], we are interested in the Majority rule Sorting method (MR-Sort)[14] which corresponds to a simplified version of the ELECTRE TRI method [8].

In line with the preference learning / disaggregation paradigm [11], we consider the case where we learn MR-Sort parameters from preference data, i.e., from assignment examples. In the literature, learning algorithms consider preference orders on criteria as given. We consider the broader case where the order is not known.

[1] MICS, CentraleSupélec, Université Paris-Saclay, Gif-sur-Yvette, France
[2] IBM, Saclay, France
[3] IBM Research – Zurich, Rüschlikon, Switzerland

When preference orders on criteria are fully unknown, learning an MR-Sort model from data becomes a highly flexible problem. We consider a simpler case in which evaluations of alternatives induce monotone preferences, but the preference directions on criteria are unknown (i.e., whether each criterion is to be maximized or minimized). Hence, we aim at uncovering from training data the criteria preference directions, and the MR-Sort parameters, simultaneously. In this paper, we design and test an algorithm that learns MR-Sort parameters with latent criteria preference directions.

The paper is organized as follows. In the first section, we review the literature that considers non-monotone preferences when learning a multicriteria model from preference data. The second and third sections recall the MR-Sort model and the inverse problem (Inv-MR-Sort), which consists in learning MR-Sort parameters from a dataset. In Section 4, we propose an algorithm for learning MR-models with latent criteria preference directions. In section 5, the proposed algorithm is numerically evaluated. A final section groups conclusions and further research.

2 Related work

In Multiple Criteria Decision Aid (MCDA), preference learning methods require a preference order on criteria. Such preference order on criteria directly results from the fact that alternatives evaluations / scores correspond to performances that are to be maximized (profit criterion) or minimized (cost criterion). This naturally results in monotone preference data. In our work, we still consider monotone preferences, but we focus on determining whether each criterion is to be maximized or minimized. We are not aware of previous work considering such case. However, related work in the literature deals with the case of non-monotone preferences.

In the context of preference learning / disaggregation, several approaches consider non-monotone preferences on criteria. To the best of our knowledge, however, all these contributions consider a utility-based preference model, in which non-monotone attributes are represented using non-monotone marginal utility functions.

Historically, Despotis and Zopounidis [5] are the first to consider single peaked value functions with an additive piece-wise linear model. The UTA-NM method proposed in [13] allows for non-monotone marginals and prevents over-fitting by introducing a shape penalization. Also in the context of an additive utility model, Eckhardt and Klieger [7] define a heuristic pre-processing technique to transform arbitrary attributes input into a space monotone w.r.t. the Decision Maker's (DM) preferences. Liu *et al.* [15] model sorting with a piecewise linear additive sorting model, using a regularization framework to limit non-monotonicity. Guo *et al.* [10] propose a progressive preference elicitation for multicriteria sorting using a utility model with

non-monotone attributes. A framework to rank alternatives with a utility model using slope variation restrictions for marginals is proposed in [9]. Based on a mixed-integer program, [12] proposes to disaggregate an additive piece-wise linear sorting model. This model uses different types of monotone (increasing, decreasing) and non-monotone (single peaked, single caved) marginal value functions. They finally illustrate the method on an application concerning nanomaterial exposure management. Recently, Sobrie et al. [18] consider single-peaked preferences when learning an MR-Sort model in a medical application.

3 The MR-Sort model

The Majority Rule Sorting method (MR-Sort) [14] is a multiple criteria method which aims at assigning alternatives evaluated on multiple criteria to one of the predefined ordered categories $C^1, \ldots, C^p$ (C^1 and C^p are the worst and best categories, respectively). We denote $\mathcal{N} = \{1, ..., n\}$ the set of n criteria, and X_i the set of possible evaluations on criterion $i \in \mathcal{N}$. An alternative a is thus represented by a tuple of evaluations $(a_1, \ldots, a_n) \in \prod_{i \in \mathcal{N}} X_i$, $a_i \in X_i$ being the evaluation of alternative a on criterion i. We denote $X = \prod_{i \in \mathcal{N}} X_i$ the cartesian product of criteria scales.

Evaluations on each criterion should either be maximized (in case of a "*profit*" criterion, i.e., the greater, the better), or minimized (in case of a "*cost*" criterion, i.e., the lower, the better). Hence, the preference order $\succcurlyeq_i \subset X_i^2$ for each criterion $i \in \mathcal{N}$ is defined as follows:

- for "*profit*" criterion, $a_i \succcurlyeq_i a_i'$ iff $a_i \geq a_i'$, $a_i, a_i' \in X_i$,
- for "*cost*" criterion, $a_i \succcurlyeq_i a_i'$ iff $a_i \leq a_i'$, $a_i, a_i' \in X_i$,

MR-Sort considers $p-1$ multicriteria limit profiles $b^1, \ldots, b^{p-1}$ separating consecutive categories, where profile $b^h = (b_1^h, .., b_n^h) \in X$ delimits the frontier between C^h and C^{h+1}. Furthermore, we denote b^0 (b^p, resp.) the lower profile of category C^1 (the upper profile of category C^p, resp.) defined such that $a_i \succcurlyeq_i b_i^0, \forall i, \forall a_i \in X_i$ (not$(a_i \succcurlyeq_i b_i^p), \forall i, \forall a_i \in X_i$, resp.).

MR-Sort assigns alternative $a \in X$ to category C^h, $h \in \{1, \ldots, p\}$ (denoted $c(a) = h$), when the set of criteria for which a_i is better than the lower profile of C^h ($a_i \succcurlyeq_i b_i^{h-1}$) forms a majority, but the set of criteria for which a_i is better than the upper profile of C^h ($a_i \succcurlyeq_i b_i^h$) is not a majority. The notion of majority is formalized using weights $w_1, ..., w_n$ attached to criteria (with $w_i \geq 0$, $\forall i$, and $\sum_{i \in \mathcal{N}} w_i = 1$), and a majority threshold $\lambda \in [0.5; 1]$: a subset of criteria $\mathcal{I} \subseteq \mathcal{N}$ is a majority iff $\sum_{i \in \mathcal{I}} w_i \geq \lambda$. Finally, the MR-Sort rule can be expressed as follows:

$$c(a) = h \quad \Leftrightarrow \quad \sum_{i: a_i \succcurlyeq_i b_i^{h-1}} w_i \geq \lambda \text{ and } \sum_{i: a_i \succcurlyeq_i b_i^h} w_i < \lambda \qquad (1)$$

The Non-Compensatory Sorting (NCS) method [3, 4] has been axiomatized in [3, 4], and MR-Sort corresponds to NCS when majorities can be represented additively, and without vetoes.

4 Learning MR-Sort model from preference data

In order to set appropriate values for the MR-Sort parameters (weights, majority level, and limit profiles), we consider a learning paradigm, in which a set of assignment examples is provided by the decision-maker; the aim is to *extend* these assignments using an MR-Sort model. To do so, we search for the MR-Sort parameters that best match the set of assignment examples.

4.1 MR-Sort and Inv-MR-Sort

In this paper, we refer to an *assignment* as a function mapping a subset of *reference alternatives* $A^\star \subset X$ to the ordered set of categories $C^1, \ldots, C^p$. These reference alternatives highlight values of interest on each criterion $i \in \mathcal{N}$, $X_i^\star = \bigcup_{x \in X^\star} \{x_i\}$. We refer to the problem of finding suitable preference parameters specifying an MR-Sort model by *Inv-MR-Sort*.

We call *learning set* a couple: $L = (A^*, C)$, where $C = (c(a), \forall a \in A^*)$; that is each alternative $a \in A^*$ is assigned to a desired category $c(a) \in C$. Hence, *Inv-MR-Sort* takes as input a learning set L and returns MR-Sort parameters (b, w, λ) that best match the learning set.

Example 4.1. *Consider the example mentioned in the introduction in which a set of clients should be sorted into two categories with respect to their interest in buying a new Windows tablet: C^1 clients which are not interested , and C^2 potential buyers of this product. In this sorting problem, we consider three criteria: Windows PC turnover (k€), Pack Office Turnover (k€), and the Linux PC turnover (k€). As previous Windows PC and Pack office purchases indicate an interest in a Windows environment, both first criteria should be maximized (profit criterion); conversely, as previous purchases of Linux PCs indicates disinterest for Windows products, the third criterion should be minimized (cost criterion). We have a learning set of five clients evaluated on all three criteria together with a desired assignment, see Table 1 below. Given this learning set, Inv-MR-Sort computes the following set of parameters: b^1=(540k€, 53k€,310k€), w=(0.55,0.12,0.33), λ=0.61.*

	Windows PC Turnover (↑)	Pack Office Turnover (↑)	Linux PC Turnover (↓)	Category
client 1	500 k€	20 k€	300 k€	C^1
client 2	200 k€	10 k€	350 k€	C^1
client 3	800 k€	90 k€	150 k€	C^2
client 4	600 k€	50 k€	300 k€	C^2
client 5	900 k€	70 k€	250 k€	C^2

Table 1: Example 1: learning set of 5 clients

Previous works have tackled the problem of learning an MR-Sort model (or a similar sorting model) from a dataset. [16] proposed a nonlinear programming formulation to learn the parameters for the ELECTRE TRI sorting method, which is an ancestor of the MR-Sort model, see [8]). Later on, [14] uses a mixed-integer linear program to solve the Inv-MR-Sort problem. These two techniques only allow for solving instances of small size due to computational difficulty.

More recently efficient Boolean Satisfiability formulations for learning NCS models from data have been proposed [2], [1]. These SAT/MaxSAT formulations make it possible to handle larger datasets. Recently, an evolutionary population-based heuristic has been proposed to solve Inv-MR-Sort, see [19], [17]. We provide hereafter a description of this heuristic, as our work takes this heuristic as a starting point.

4.2 Heuristic for Inv-MR-Sort

The heuristic proposed in [17, 19] is an evolutionary population-based algorithm and learns an MR-Sort model that best matches a learning set composed of assignment examples. Each individual in the population is an MR-Sort model, i.e., values for limit profiles b^h, criteria weights w_i, and the majority level λ; we denote each individual by

$(\langle b \rangle, w, \lambda)$. After an initialization step which generates a first population, the algorithm proceeds to evolving the population of MR-Sort models iteratively until a model in the population perfectly restores the learning set, or a maximum number of max_{it} iterations is reached.

At each iteration, the algorithm tries to improve the fitness of each MR-Sort model in the population (fitness corresponds to classification accuracy (CA), i.e., the proportion of correctly restored examples in the learning set) by performing two consecutive steps: (*i*) optimize the weights and majority level (limit profiles being fixed) using linear programming (LP), and (*ii*) improve heuristically the limit profiles (weights and majority level being fixed). The 50% best models are kept in the population for the next iteration, while 50% new MR-Sort models are randomly generated. We summarize below in Algorithm 1 the main steps of the heuristic as well as its inputs/outputs.

Algorithm 1: Inv-MR-Sort heuristic, [19]

Input: L: learning set
Output: model $(\langle b \rangle, w, \lambda)$ that best match L in the population
$it \leftarrow 1$
Initialize POP, a population of n_{mod} models
while *($it \leq max_{it}$) and (no model in POP fully restores L)* **do**
  foreach *model* $(\langle b \rangle, w, \lambda) \in POP$ **do**
    Optimize weights w and threshold λ using LP
    Improve profiles $\langle b \rangle$ heuristically
  end
  Renew the $\lfloor n_{mod}/2 \rfloor$ worst models in POP
  $it \leftarrow it + 1$
end
return $(\langle b \rangle, w, \lambda)$ that best match L in POP

5 Learning an MR-model with latent criteria preference directions

The heuristic described by [17, 19] assumes the monotonicity of criteria in the MR-Sort model to be learned. In [17], the definition of the Inv-MR-Sort problem assumes, without loss of generality, that the decision-maker (DM) preferences are increasing with the criteria performances (the greater, the better). In this work, we aim at extending the Inv-MR-Sort problem to the case where preferences are still monotone, but the criteria preference directions are not known, i.e., we do not know whether criteria are to be maximized or minimized.

Hence, we aim at learning from a learning set, criteria weights w, the majority threshold λ, the profiles $\langle b \rangle$, and the criteria preference direction (at least for one of them). Example 5.1 extends Example 4.1, and illustrates this learning situation when the preference direction is to be learned only for one criterion.

Example 5.1 (Example 4.1 cont.)**.** *Pursuing example 4.1, we add a criterion "Dual boot PC Turnover" considered as important by the DM. It evaluates the dual-boot PC purchases (in k€) for a given client. As mentioned before, the preference direction of this criterion is not known a priori. We report in Table 2 the performance table with the additional latent criteria (Dual boot PC Turnover), as well as the classification of the 5 clients.*

5.1 How to learn preference directions

Consider a situation (similar to example 5.1) in which all criteria directions are known except for criterion i. Suppose we use Algorithm 1 to learn an MR-Sort model $(\langle b \rangle, w, \lambda)$ from a dataset, hypothesizing incorrectly the preference direction for criterion i (supposing criterion i is to be maximized when the ground truth states that is should be minimized, or vice versa). In such a case, Algorithm 1 will favor models that inhibited criterion i so as to best restore the learning set. Models that inhibit criterion i are those for which $w_i = 0$, or the ones for which $b_i^h > Max_{a \in A^*}\{a_i\},\ \forall h$ or $b_i^h < Min_{a \in A^*}\{a_i\},\ \forall h$.

Indeed, if Algorithm 1 returns a model $(\langle b \rangle, w, \lambda)$ with a weight w_i close to zero, or profiles $\langle b \rangle$ such that b_i^h are close to the endpoints of the scale X_i, this is a strong sign that the direction of preference could be incorrectly hypothesized. Such consideration will be useful to define the algorithm proposed in §5.3.

5.2 Inv-MR-Sort problem with latent criteria preference directions

The preference direction d_i describes how the preference relation $\succcurlyeq_i$ on criterion i relates to the evaluations on the criterion scale X_i. Criterion i has an increasing (decreasing, resp.) preference direction, noted $d_i = 1$ ($d_i = -1$, resp.), when criterion i is to be maximized (minimized, resp.), i.e. is a profit (cost, resp.) criterion. The vector of criteria preference directions is noted: $d = \{d_1, ..., d_n\}$; d can be considered as another parameter of the MR-Sort model. In our paper, a criterion whose preference direction is not known (and yet to be learned) is called a *criterion with latent preference direction*.

Therefore, we extend the Inv-MR-Sort problem to a broader problem that encompasses the learning of limit profiles, weights and majority thresholds, together with the preference directions of latent criteria. We denote it $IMS_{q|n}$, the *Inv-MR-Sort* problem that aims at learning q preference directions over n criteria in the model ($q \leq n$). We call $\mathcal{Q}$ the set of latent preference direction criteria ($\mathcal{Q} \subseteq \mathcal{N}$ and $|\mathcal{Q}| = q$). In the following, we consider the $IMS_{q|n}$ problem, that aims at inferring the tuple of parameters $(b, w, \lambda, \{d_i : \forall i \in \mathcal{Q}\})$.

5.3 An algorithm for Inv-MR-Sort with latent criteria preference directions

We present in this section, a two-step method to solve $IMS_{q|n}$ (Inv-MR-Sort with q latent criteria preference directions). The two consecutive steps consist first in *(i)* learning the unknown preference directions, and then *(ii)* learning other parameters $(w, \langle b \rangle, \lambda)$.

In the first step, Algorithm 1 is executed on a modified version of the $IMS_{q|n}$ in which latent criteria are duplicated, yielding a problem that takes into account the two types of preference directions for the q latent criteria. This allows to induce the appropriate preference directions for the q latent criteria. In the second step, Algorithm 1 is performed with the q preference directions that are fixed, in order to determine the remaining parameters of the MR-Sort model.

5.3.1 The first stage

The first stage performs consecutively:

- the transformation of $IMS_{q|n}$ to an intermediate problem ($IMS_{0|n+q}$) obtained through the duplication of criteria with unknown preference directions,
- the resolution of $IMS_{0|n+q}$ with Algorithm 1.
- the deduction of the q preference directions of the initial problem from the outcome of $IMS_{0|n+q}$

	Windows PC Turnover (↑)	Pack Office Turnover (↑)	Linux PC Turnover (↓)	Dual-boot PC Turnover (?)	$c(a)$
client 1	500 k€	20 k€	300 k€	150 k€	1
client 2	200 k€	10 k€	350 k€	130 k€	1
client 3	800 k€	90 k€	150 k€	100 k€	2
client 4	600 k€	50 k€	300 k€	100 k€	2
client 5	900 k€	70 k€	250 k€	100 k€	2

Table 2: Assignment examples: dataset of 5 clients and 4 criteria with one unknown preference direction (Dual-boot PC turnover)

From $IMS_{q|n}$ to $IMS_{0|n+q}$: Considering $IMS_{q|n}$, we duplicate the subset $\mathcal{Q}$ into a similar set $\mathcal{Q}'$; the preference directions of criteria in $\mathcal{Q}'$ and $\mathcal{Q}$ are opposite. More precisely, w.l.o.g., we choose to set an increasing preference direction ($d_i = 1$) to criteria in $\mathcal{Q}$, and a decreasing preference direction ($d_i = -1$) to criteria in $\mathcal{Q}'$. In problem $IMS_{0|n+q}$, alternatives in the learning set have the same evaluations for criteria in $\mathcal{Q}'$ and in $\mathcal{Q}$.

Thus, we associate to each criterion $i \in \mathcal{Q}$ a criteria $i' \in \mathcal{Q}'$ that shares the same performance values over the alternatives: they both account for the same initial criterion whose preference direction is unknown. We call $\mathcal{AC}$ the set of such pairs of criteria (i, i'). This modified problem thus obtained can be translated as $IMS_{0|n+q}$, since there are no more unknown preference directions and the total number of criteria rises to $n + q$. The intuition behind the duplication of i to i' is to foster the algorithm to inhibit the criterion with the "incorrect" preference direction, while making the other criterion more influential.

Resolution of $IMS_{0|n+q}$: At this step, we solve $IMS_{0|n+q}$ using Algorithm 1. The problem comprises $n + q$ criteria which implies the learning of $n + q$ weights, profiles $\langle b \rangle$ of dimension $n + q$, as well as a threshold λ.

Deduction of the q preference directions: After the resolution of $IMS_{0|n+q}$, we interpret the preference directions of the q latent criteria. Our reasoning is the following. Considering the resulted parameters $(w, \langle b \rangle, \lambda)$ of $IMS_{0|n+q}$:

- $\forall\ (i, i') \in \mathcal{AC}$, if $w_i = 0$ and $w_{i'} \neq 0$, then we conclude that the correct criterion is the cost criterion i', since i is inhibited in the model ($w_i = 0$).
- $\forall\ (i, i') \in \mathcal{AC}$, if $w_i \neq 0$ and $w_{i'} = 0$, then we conclude that the correct criterion is the profit criterion i, since i' is inhibited in the model ($w_{i'} = 0$).
- $\forall\ (i, i') \in \mathcal{AC}$, if $w_i \neq 0$ and $w_{i'} \neq 0$, we ground our analysis on the position of profiles $\langle b \rangle$ on criteria i and i'. As mentioned in §5.1, profiles on criterion i (or i') close to the end points of the scale X_i (or $X_{i'}$) indicates that criterion i (or i') is "inhibited". Therefore, we will select the preference direction corresponding to criterion i or i' as the one for which the profile is the further away from the endpoints of the scales X_i and $X_{i'}$.
  Consider the simplest case with 2 categories, $p = 2$ (the case with more than two categories $p > 2$ can be considered at the cost of additional technicalities). Let us consider the bi-partition of the set $X_i^\star = X_i^{\star 1} \cup X_i^{\star 2}$ ($X_{i'}^\star = X_{i'}^{\star 1} \cup X_{i'}^{\star 2}$, respectively) such that $X_i^{\star 1} = \{x_i \in X_i^\star : x_i < b_i^1\}$ and $X_i^{\star 2} = \{x_i \in X_i^\star : x_i \geq b_i^1\}$ ($X_{i'}^{\star 1}$ and $X_{i'}^{\star 2}$ are defined analogously). Let us denote $\mu_i^2 = \frac{|X_i^{\star 2}|}{|X_i^\star|}$ the proportion of alternatives in the learning set whose evaluation on criterion i is greater than the profile. We define $\mu_{i'}^2$ similarly. Depending on μ_i^2 and $\mu_{i'}^2$ and a threshold $\mu \in [0, 1]$ (in our test we consider $\mu = 0.9$), we consider three cases:
  - If $\mu_i^2 \in [\mu, 1 - \mu]$ and $\mu_{i'}^2 \notin [\mu, 1 - \mu]$, criterion i has a greater discriminative power than criterion i', and therefore corresponds to the correct preference direction.
  - If $\mu_i^2 \notin [\mu, 1 - \mu]$ and $\mu_{i'}^2 \in [\mu, 1 - \mu]$, criterion i' has a greater discriminative power than criterion i, and therefore corresponds to the correct preference direction.
  - Otherwise, we apply an *ad hoc* heuristic which computes, for each pair $(i, i') \in \mathcal{AC}$ the best classification accuracy on the learning set, when considering criterion i and i' as dictator, and selects the preference direction that performs the best.
- $\forall\ (i, i') \in \mathcal{AC}$, if $w_i = w_{i'} = 0$, we apply the same *ad hoc* heuristic as above which consider i and i' as possible dictator.

5.3.2 *The second stage*

Once the q preference directions have been determined, we can reduce $IMS_{0|n+q}$ to $IMS_{0|n}$ by preserving the q right criteria previously derived and by removing their associated criteria. Next, we solve the standard $IMS_{0|n}$ problem to be solved by Algorithm 1. The resulting parameters of this last step give the remaining part of the solution to the initial problem $IMS_{q|n}$. Hence, we obtain all parameters $(w, b, \lambda, \{d_i : i \in \mathcal{Q}\})$ of the *Inv-MR-Sort* problem with latent preference direction criteria.

6 Experimental validation of the algorithm

This section presents numerical results which makes it possible to analyze the behavior of the proposed algorithm. The experiments should provide insights so as to answer the following questions:

- Regarding the computing time, how does the algorithm cope with large datasets ?
- What is the ability of the algorithms to restore an existing dataset when criteria preference directions are latent ?
- How many assignment examples should the learning set contain so that the learned model accurately classifies new alternatives ?
- How does the algorithm cope with noisy datasets ?

6.1 Experimental design

We ran our experiments on a machine endowed with Ubuntu 18.04.4 LTS (64 bits) with an Intel(R) Xeon(R) Gold 6248 CPU @ 2.5GHz and 376 GB of RAM.

In our study, we consider as an input two specific random datasets generated in accordance with a pre-constructed MR-Sort model : datasets with noise - which could be assimilated to real preferences data - and noise-free datasets. By this way, we can judge on a reliable basis the restoration of models. We call this ground truth model, M^0.

Once M^0 is constructed, we construct L by classifying randomly generated alternatives according to M^0 and in such a manner to obtain a balanced distribution of alternatives over categories. Besides, we choose not to disclose the preference directions of q out of n criteria.

In our experiments, we use 100 samples of MR-Sort M^0 in order to obtain mean values as performances of learned models in terms of computation time, restoration rate of the learning set/test set and restoration rate of the preference directions.

We uniformly generate $p-1$ random values in [0,1] for the profiles values and order them in the following order $b_i^1 \leq b_i^2 ... \leq b_i^{p-1}, \forall i \in \mathcal{N}$. The performance values of A^* are also generated uniformly in [0,1]. In order to assign values to weights, we first draw $|\mathcal{N}|-1$ numbers in [0,1]. These numbers added to 0 and 1 are ranked in ascending order. The difference between each successive number pairs in the ranking forms the values of criteria weights. We randomly generate λ in]0.5,1[.

The experimental parameters and their possible values taken into account for the generation of a dataset are : the number of criteria $|\mathcal{N}|$ in $\{5, 7, 9, 11\}$, the number of latent criteria q in $\{1, 2, \ldots, |\mathcal{N}|\}$, the number of categories p in $\{2, 3, 4, 5\}$, the learning set size $|A^*|$ in $\{100, 250, 500, 750, 1000, 1250, 1500\}$, and noise rate σ in $\{0, 0.05, 0.1, 0.15, 0.2, 0.25\}$. Concerning the generation of noisy data, the process is the following. First, we generate a learning set from M^0. Then we randomly choose $\sigma \times |A^*|$ examples of the learning set that will be subject to error. For these alternatives, we change their assignment to an adjacent category.

We execute our approach as described in Section 5 in order to solve $IMS_{q|n}$. M^0 represents a ground truth on which the resulting models of the approaches are validated. We expect a good rendering of solutions since M^0 comes from a real MR-Sort model, at least with noise-free datasets.

We define 4 measures in order to evaluate the quality of the restored models : the execution time, the classification accuracy CA of the learning set, the classification accuracy CA on the test set. The latter is measured on a dataset of 10000 alternatives. We also consider two indicators in order to appreciate the restoration of preference directions : PDR1 which is the rate of restoring all the preference directions at once, while PDR2 is the proportion of restoring one preference direction on average.

6.2 Results

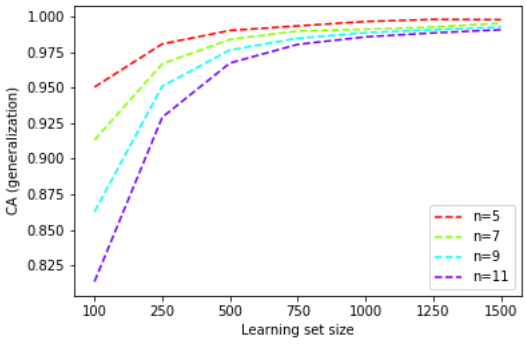


Figure 1: Classification Accuracy (CA) of the test set (generalization) per number of criteria (n) and learning set size for a problem involving 1 latent criterion, 2 categories and noisy-free learning sets

6.2.1 *Execution time and memory requirements*

The execution time of the approach increases in function of 4 parameters : the learning set size, the number of criteria n, the number of latent criteria q and the number of categories p. As an example, the execution time on a laptop rises to 545 seconds when considering the problem $IMS_{1|11}$, with 5 categories and 1500 alternatives in the learning set without noise. The memory space required is less than 50MB.

6.2.2 *Varying the number of criteria (n)*

First, our approach was executed for a series of problems involving different criteria with 1 latent criterion, 2 categories and considering noisy-free data sets. The ability of the algorithm to restore assignment examples diminishes with the increase of the number of criteria in the problem (Figure 1). However the classification accuracy surges to 95% with 500 alternatives in the learning set considering the problem $IMS_{1|11}$ and then converges towards 1.

6.2.3 *Varying the number of criteria with latent preference directions (q)*

Second, we tested our approach on instances of 7 criteria, 2 categories, with noisy-free data sets and varied the number of latent criteria. The classification accuracy of the learning set is flawless (ranges between 99% and 100%) regardless the number of latent criteria and the learning set size. The classification accuracy in generalization converges towards 1 regardless the number of latent criteria (Figure 2a). The algorithm behaves exactly in the same manner independently of the number of latent criteria, i.e without additional difficulties to restore assignments. With only 250 alternatives in the learning set, 96% of new assignments are restored. The PDR1 increases with the size of the learning set and converges differently depending of the number of latent criteria, towards 1 (Figure 2b). Nevertheless, the algorithm behaves in general better with less latent criteria. With 750 alternatives in the learning set, the algorithm performs with a CA more than 85% considering 7 latent criteria. This is by far better than the random probability of retrieving the preference directions of 7 latent criteria that is $1/2^7$ ($\approx 0{,}8\%$).

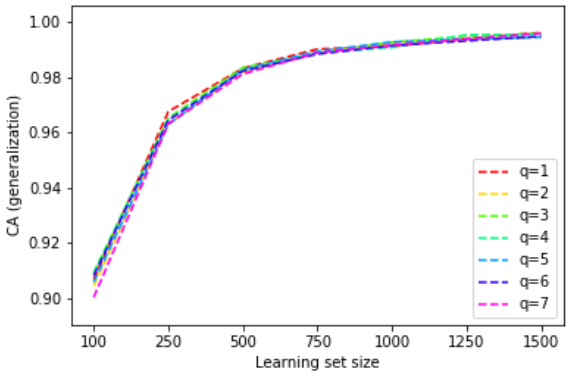
6.2.4 *Varying the number of categories*

Third, we tested our approach on instances of 7 criteria, 7 latent criteria, with noisy-free data sets and varied number of categories. The classification accuracy decreases progressively according to the increase of the number of categories in the problem (Figure 3a). Despite this, the CA (in generalization) still increases when the learning size set becomes greater. With 5 categories, and 500 assignment examples in the learning set, the algorithm is able to restore more than 90% of new assignment examples.

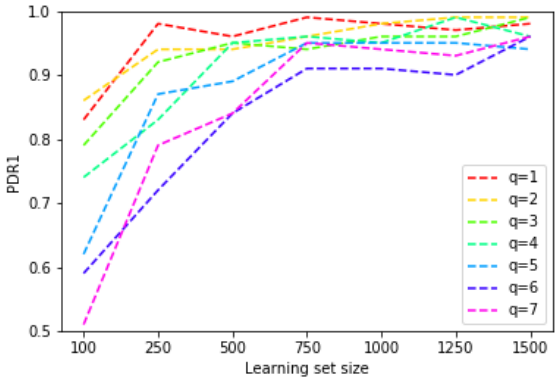
The preference direction restoration degrades moderately with more than 2 categories, but increases with the size of the learning set (Figure 3b). The PDR1 reaches at least 80% with more than 750 alternatives in the learning set regardless the number of categories.

6.2.5 *Considering noisy learning sets*

The fourth test of our approach concerns the case with 7 criteria, 7 latent criteria, 2 categories and considering the presence of the noise (σ) in the learning set. The figure 4a teaches us the ability of our approach to restore more than $100*(1-\sigma)\%$ of new assignments examples, which means that the algorithm is fairly robust. In addition, the CA (in generalization) still increases proportionally to the presence

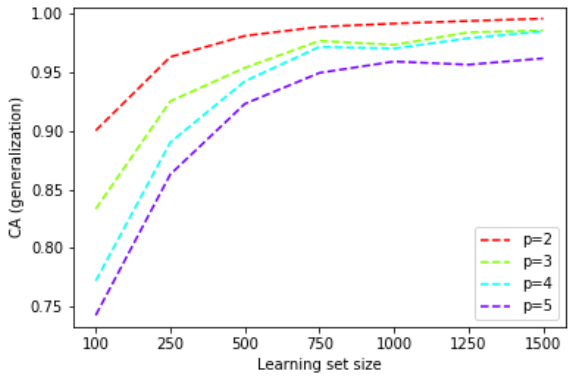


(a) Classification accuracy of the test set (generalization)

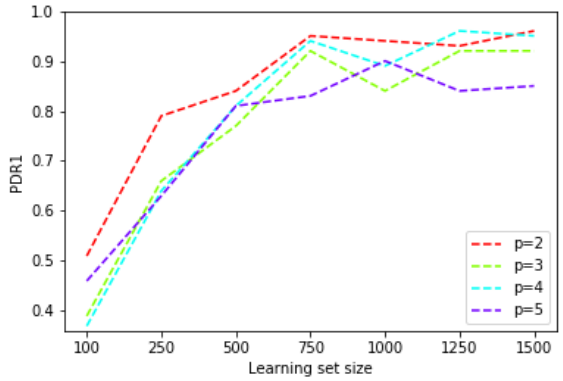


(b) Preference direction restoration (PDR1)

Figure 2: Results for problems involving 7 criteria, 2 categories and noisy-free learning sets per number of latent criteria q and learning set size



(a) Classification accuracy of the test set (generalization)



(b) Preference direction restoration (PDR1)

Figure 3: Results for problems involving 7 criteria, 7 latent criteria and noisy-free learning sets per number of categories p‘ and learning set size

of errors. As an example, the CA attained 85% with 25% of noise when the learning set size is 500.

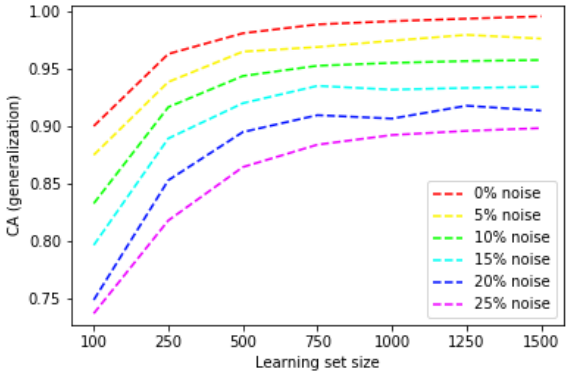
The figure 4b shows us that the impact of more noise is not so strong on the ability of the algorithm to restore preference directions. The PDR2 increases with difficulty along with the learning set size once in the presence of noise.

6.3 Discussion

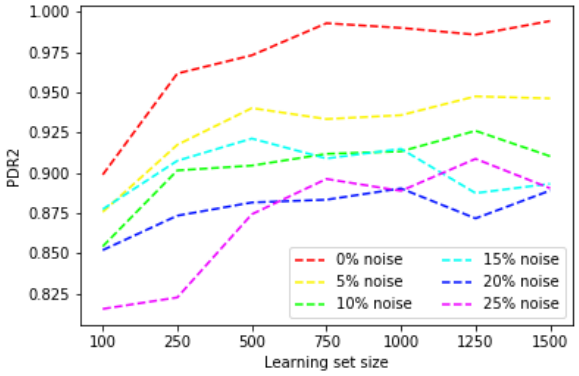
Useful insights have been found through these experiments. First, it is apparent that our algorithm succeeds in retrieving preference directions as well as the other MR-Sort parameters of our problem, considering noisy-free learning sets and 2 categories. Indeed, the classification accuracy of the learning set is excellent ($\approx 99\%$) regardless of the number of latent criteria and the number of criteria considered in our tests ($n \in \{5, 7, 9, 11\}$). In generalization, the behaviour of the algorithm is similar : as a matter of fact, the restoration of new assignments is as good as the restoration of the Sobrie's problem (which is $IMS_{0|n}$) independently of the number of latent criteria and the learning set size. The approach also succeeds in restoring preference directions since the restoration converges quite quickly towards 1 (with $|A^\star| = 1500$). Moreover, additional results show that PDR2 - which considers the restoration of one preference direction on average - reaches 95% for $IMS_{q|7}$, with $0 \leq q \leq 7$ with only 250 assignment examples in the learning set.

Second, the computational time is fairly affordable since it is close to the Sobrie's heuristic. The worst-case scenario considered in our tests (which is the learning of $IMS_{7|7}$ with $|A^\star| = 1500$ and 4 categories considering 25% of noise in the learning set) indicates that the algorithm runs less than 15 minutes and the memory space needed is less than 50MB (using a machine endowed with 2,3 GHz Intel Core i5 and 8Go of RAM). Indeed the execution time strongly depends on the number of iterations of the outer loop of the algorithm ; it is therefore controlled by design.

Unsurprisingly, considering the context without noise, the greater the number of criteria and the number of categories, the lower is the classification accuracy for a given learning set size and regardless the number of latent criteria. For illustration, 100 alternatives in the learning set is sufficient to restore 95% in generalization with a 5-criteria model and 2 categories, while to get the same restoration rate, we need 500 alternatives for a 11-criteria model. Analogously, less than 250 alternatives in the learning is responsible for a classification accuracy at 95% in generalization for a 7-criteria model with 2 categories while a learning set of 1000 alternatives is needed for the same CA score and for a model with 5 categories. Therefore, it is possible to obtain a good performance as long as we provide a large learning set.



(a) Classification Accuracy (CA) of the test set (generalization) per noise percentage and learning set size



(b) Preference direction restoration (PDR2) per number of noise percentage and learning set size

Figure 4: Results of our approach for a problem involving 7 criteria, 7 latent criteria, 2 categories

As expected in the presence of noisy data in the learning set, the classification accuracy in generalization is reduced in function of the increase of the noise introduced in the learning set. This is all the more linked with the difficulty to learn the preference direction (observed in figure 4b) as soon as we deal with noisy data in comparison to the case without noise (in red line). Nevertheless, our approach takes advantage of the increase of the learning set size since it still succeeds at restoring more assignments (CA in generalization) with more assignment examples in the learning set size.

Conclusion and Perspectives

In this paper we have considered the MR-Sort model in the case where the direction of preference of criteria is unknown. We have proposed a solution which extends the heuristic method introduced by Sobrie *et al.* to this case. For a given learning set, our algorithm estimates the unknown preference directions as well as the parameters of the MR-Sort model from the learning set.

Extensive numerical simulations demonstrate the capability of the algorithm to correctly estimate both preference directions and the other model parameters with an accuracy over 90 % (for a noisy-free learning set of 250 examples). Moreover, the algorithm showed to be robust in the case of noisy data. Finally, the proposed solution features a very contained computational complexity both in training and inference phases. All these characteristics make this approach very suitable for real-world applications.

As future work, we will extend this approach to other non-monotone preferences, namely the case of single peaked preferences.

REFERENCES

[1] K. Belahcène, *Towards accountable decision aiding : explanations for the aggregation of preferences*, Ph.D. dissertation, CentraleSupélec, Université Paris-Saclay, 2018.

[2] K. Belahcène, C. Labreuche, N. Maudet, V. Mousseau, and W. Ouerdane, 'An efficient SAT formulation for learning multiple criteria non-compensatory sorting rules from examples', *Computers & Operations Research*, **97**, 58–71, (2018).

[3] D. Bouyssou and T. Marchant, 'An axiomatic approach to noncompensatory sorting methods in mcdm, i: The case of two categories', *European Journal of Operational Research*, **178**, 217–245, (02 2007).

[4] D. Bouyssou and T. Marchant, 'An axiomatic approach to noncompensatory sorting methods in MCDM, II: More than two categories', *European Journal of Operational Research*, **178**(1), 246–276, (April 2007).

[5] D. K. Despotis and C. Zopounidis, *Building Additive Utilities in the Presence of Non-Monotonic Preferences*, 101–114, Springer, 1995.

[6] M. Doumpos and C. Zopounidis, *Multicriteria Decision Aid Classification Methods*, 2002. Springer.

[7] A. Eckhardt and T. Kliegr, 'Preprocessing algorithm for handling non-monotone attributes in the UTA method', in *Proceedings of the ECAI-12 Workshop on Preference Learning: Problems and Applications in AI (PL-12)*, eds., J. Fürnkranz and E. Hüllermeier, (2012).

[8] J. Figueira, V. Mousseau, and B. Roy, 'Electre methods', in *Multiple criteria decision analysis: State of the art surveys*, 133–153, Springer, (2005).

[9] M. Ghaderi, F. Ruiz, and N. Agell, 'A linear programming approach for learning non-monotonic additive value functions in multiple criteria decision aiding', *European Journal of Operational Research*, **259**(3), 1073 – 1084, (2017).

[10] M. Guo, X. Liao, and J. Liu, 'A progressive sorting approach for multiple criteria decision aiding in the presence of non-monotonic preferences', *Expert Systems with Applications*, **123**, 1 – 17, (2019).

[11] E. Jacquet-Lagrèze and Y. Siskos, 'Preference disaggregation: 20 years of MCDA experience', *European Journal of Operational Research*, **130**(2), 233–245, (April 2001).

[12] M. Kadziński, K. Martyn, M. Cinelli, R. Słowiński, S. Corrente, and S. Greco, 'Preference disaggregation for multiple criteria sorting with partial monotonicity constraints: Application to exposure management of nanomaterials', *International Journal of Approximate Reasoning*, **117**, 60 – 80, (2020).

[13] T. Kliegr, 'UTA-NM : Explaining stated preferences with additive non-monotonic utility functions', in *Proceedings of ECML PKDD Workshop on Preference Learning*, (2009).

[14] A. Leroy, V. Mousseau, and M. Pirlot, 'Learning the parameters of a multiple criteria sorting method', pp. 219–233, (10 2011).

[15] J. Liu, X. Liao, M. Kadziński, and R. Słowiński, 'Preference disaggregation within the regularization framework for sorting problems with multiple potentially non-monotonic criteria', *European Journal of Operational Research*, **276**(3), 1071 – 1089, (2019).

[16] Vincent Mousseau and Roman Slowinski, 'Inferring an ELECTRE TRI model from assignment examples', *Journal of Global Optimization*, **12**(2), (1998).

[17] O. Sobrie, *Learning preferences with multiple-criteria models*, Ph.D. dissertation, Université de Mons (Faculté Polytechnique) and Université Paris-Saclay (CentraleSupélec), June 2016.

[18] O. Sobrie, Said Lazouni, M., V. Mousseau, and M. Pirlot, 'A new decision support model for preanesthetic evaluation', *Computer Methods and Programs in Biomedicine*, **133**, 183–193, (2016).

[19] O. Sobrie, V. Mousseau, and M. Pirlot, 'Learning monotone preferences using a majority rule sorting model', *International Transactions in Operational Research*, **26**(5), 1786–1809, (2019).