



HAL
open science

Interactive Elicitation of a Majority Rule Sorting Model with Maximum Margin Optimization

Ons Nefla, Meltem Öztürk, Paolo Viappiani, Imène Brigui-Chtioui

► **To cite this version:**

Ons Nefla, Meltem Öztürk, Paolo Viappiani, Imène Brigui-Chtioui. Interactive Elicitation of a Majority Rule Sorting Model with Maximum Margin Optimization. ADT 2019 - 6th International Conference on Algorithmic Decision Theory, Oct 2019, Durham, NC, United States. hal-02285040

HAL Id: hal-02285040

<https://hal.science/hal-02285040v1>

Submitted on 12 Sep 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Interactive Elicitation of a Majority Rule Sorting Model with Maximum Margin Optimization

Ons Nefla^{1,2}, Meltem Öztürk¹, Paolo Viappiani³, and Imène Brigui-Chtioui²

¹ Université de Paris Dauphine, PSL Research University CNRS, LAMSADE, 75016 Paris, France,

ons.nefla@dauphine.eu

meltem.ozturk@dauphine.fr

² Emlyon business school, 23 Avenue Guy de Collongue, Ecully, France

brigui-cthioui@em-lyon.com

³ Sorbonne Université, CNRS, Laboratoire d’informatique de Paris 6, LIP6, F-75005 Paris, France

paolo.viappiani@lip6.fr

Abstract. We consider the problem of eliciting a model for ordered classification. In particular, we consider *Majority Rule Sorting (MR-sort)*, a popular model for multiple criteria decision analysis, based on pairwise comparisons between alternatives and idealized profiles representing the “limit” of each category.

Our interactive elicitation protocol asks, at each step, the decision maker to classify an alternative; these assignments are used as training set for learning the model. Since we wish to limit the cognitive burden of elicitation, we aim at asking informative questions in order to find a good approximation of the optimal classification in a limited number of elicitation steps. We propose efficient strategies for computing the next question and show how its computation can be formulated as a linear program. We present experimental results showing the effectiveness of our approach.

Keywords: preference elicitation · ordinal classification · incremental elicitation · MR-sort · simulations.

1 Introduction

There are several situations where it is necessary to classify objects, defined on several criteria, into ordered classes (for example, credit ratings, evaluating students, hotel categorization, etc). Such ordinal classification problems, also called *multi-criteria sorting*, have been considered by ELECTRE TRI [16,19], a popular method from the field of multi-criteria decision analysis that has been successfully applied to several domains.

Majority Rule Sorting (MR-sort) [15] is a simplified version of ELECTRE TRI belonging to a class of “non compensatory” decision models that have been axiomatized by Bouyssou and Marchant [5]. The remarkable characteristic of the MR-sort procedure is that its classifications are readily explainable to the users. In MR-sort an alternative is assigned to a category when it is “at least as good as” an idealized profile describing the category’s lower “limit” and “not better” than the category’s upper limit. The

limit profiles encode the characteristics that an alternative should have in order to be assigned to a particular class according to the decision maker (for example: a 4-stars hotel should have rooms of at least a certain size, it should have a swimming pool, a 5 star hotel should have a classy restaurant, etc). The relation comparing alternatives and limit profiles is based on a weighted majority rule (in Sobrie et al. [21] a non linear model for combining the weights is considered).

In order to use MR-sort in practice, it is necessary to undertake a preference elicitation phase in order to assess the parameters of the model. The “classical” approach to elicitation consists in a fine grained assessment of the model parameters; however since it is often not reasonable to directly ask the decision maker about the values of the parameters of the model, we adopt an incremental elicitation approach and we assume that the decision maker can easily classify one given alternative into one of the categories; the information provided by the user can be used to update the model and provide a better result. The intuition is that, by selectively choosing which items the decision maker could classify, we could retrieve a good approximation of the “correct” model of the decision maker (DM) even with a small number of examples. We also stress that the information provided by the user may be noisy, as she may have made a mistake in assessing the right category for the alternative.

In this paper we adopt max-margin optimization, whose main principle is the following. The currently known assignments of items to categories are encoded by a set of inequalities on the feasible parameters; a shared non-negative margin is introduced as a decision variable that is maximized in the objective function. Noisy feedback is addressed by relaxing the constraints using slack variables and adding a penalty term in the objective function for violated constraints. In this paper, we propose an incremental approach for eliciting the parameters of a MR-sort model; at each step of the elicitation procedure, the system asks a question to the user and the question is specifically chosen in order to be as “informative” as possible. Our main algorithm repeatedly uses the max-margin optimization routine with two goals 1) to make an estimation of the model parameters given the current information 2) to determine a measure of uncertainty associated with a potential question. For the second goal, we consider different strategies based on computing the maximum-margin a posteriori (considering the different answers that the decision maker could give).

We now provide a brief discussion about related works. Reasoning with preferences [7] is an important issue in artificial intelligence. Several researchers have dealt with the problem of dealing with an incompletely specified preference model and with the issue of preference elicitation [4,6,24,27]. In the machine learning community, approaches for *preference learning* have been proposed [10], including approaches for ordinal classification (see Chapter 8 in [17], and, for instance [9] and [11]).

Our work is stimulated by recently proposed approaches for eliciting multi-attribute utility functions using maximum-margin optimization [25,26] in configuration problems. The maximum-margin optimization that we adopt has been used (with some variations) in previous works [21,22] that tackled the problem of learning the weights of a MR-sort model; these works, however, did not consider incremental elicitation. Bennabou et al. [2,3] recently proposed interactive elicitation methods based on mini-max regret for ordinal classification problems (with a related but different model); these

methods however have the inconvenience that they are not tolerant to errors in the responses of the decision maker.

The paper is structured as follows. After some background in Section 2, we present MR-sort model in Section 3 and the interactive elicitation in Section 4. We then discuss our experimental results in Section 5 and provide some final comments and directions for future work in Section 6.

2 Background

We consider a set X of m items that are evaluated with respect to a *set of criteria* $\mathcal{C} = \{criterion_1, \dots, criterion_n\}$. Criteria associate items with a performance evaluation; (with little abuse of notation, we use x to refer to both the item and its evaluation vector). The *evaluation of x* is a vector $(x_1, x_2, \dots, x_n) \in \mathcal{E}_1 \times \dots \times \mathcal{E}_n$. The sets $\mathcal{E}_1, \dots, \mathcal{E}_n$ are totally ordered and represent the space of criteria evaluations. Indeed we are interested in ordinal classification methods that allow the criteria evaluations to be defined on scales not necessarily numerical, and that can differ among the criteria. We use $[m]$ to denote the set $\{1, \dots, m\}$.

MR-sort [15] is a multi-criteria ordinal classification method allowing to assign alternatives to ordered categories. The *set of categories* is denoted by $\mathcal{C} = \{C_1, C_2, \dots, C_p\}$. Categories are ordinal, C_1 being the worst and C_p the best one. Each category C_h is characterized by two “fictitious” items $b^h = (b_1^h, \dots, b_n^h) \in \mathcal{E}_1 \times \dots \times \mathcal{E}_n$ and $b^{h+1} = (b_1^{h+1}, \dots, b_n^{h+1}) \in \mathcal{E}_1 \times \dots \times \mathcal{E}_n$; these are called the *limit profiles of C_h* and we denote by $B = \{b^1, b^2, \dots, b^{p+1}\}$ the set of such limit profiles. Limit profiles play the role of the lower and the upper bounds of the category C_h ; limit profiles of higher categories dominate the lower ones: $\forall k = 1, \dots, n, b_k^{i+1} \geq_k b_k^i$ and $\exists j, b_j^{i+1} >_k b_j^i$, where \geq_k is the binary relation on the evaluations on the *criterion $_k$* . There are two special limit profiles, b^1 and b^{p+1} that are defined as the minimum and the maximum values: $b^1 = (b_1^1, \dots, b_n^1)$, such that $\forall i, b_i^1 = \min_{x \in X}(x_i)$ (resp. $b^{p+1} = (b_1^{p+1}, \dots, b_n^{p+1})$, such that $b_i^{p+1} = \max_{x \in X}(x_i)$). To fully specify a MR-sort model we need to associate each criterion cr_i with a numerical weight w_i , that intuitively represents its importance. A parameter λ is called the *majority threshold*, whose role will become clear below.

The procedure for making assignments is based on pairwise comparisons between objects of X and limit profiles. An alternative x is assigned to the category C_h if it is “at least as good as” the lower limit profile b^h and it is not “at least as good as” the upper limit profile b^{h+1} according to a binary relation \succeq . Indifference : $x \sim y \iff x \succeq y$ and $y \succeq x$.

$$x \rightarrow C_h \iff x \succeq b^h \text{ and } x \not\succeq b^{h+1} \quad (1)$$

where $x \rightarrow C_h$ means that alternative x is assigned to category C_h . An item x that is indifferent to the upper limit profile of the best category ($x \sim b^{p+1}$, meaning $x \succeq b^{p+1}$ and $x \preceq b^{p+1}$) is assigned to C_p . The binary relation $\succeq \subseteq (\mathcal{E}_1 \times \dots \times \mathcal{E}_n) \times (\mathcal{E}_1 \times \dots \times \mathcal{E}_n)$ is based on the *weighted majority principle*:

$$x \succeq b^h \iff \sum_{i: x_i \geq_i b_i^h} w_i \geq \lambda \quad (2)$$

Example. Assume that we have 5 hotels (x, y, z, t, u) defined on $n = 3$ criteria (cost, comfort, quality of the restaurant). Our aim is to assign our hotels into $p = 3$ categories (C_1 : 1 star, C_2 : 2 stars, C_3 : 3 stars). The MR-sort model parameters are: $\lambda = 0.55$, $w = (0.2, 0.5, 0.3)$. The criteria evaluation scale is between 0 and 5 (5 being the best score) for $crit_{1}$ and $crit_{2}$; the scale for $crit_{3}$ is between 0 and 10. Table 1 presents the performances of the limit profiles and the alternatives; Table 2 their comparisons using \succeq and the assignments of alternatives.

	$crit_{1}$	$crit_{2}$	$crit_{3}$
b^1	0	0	0
b^2	2	2	4
b^3	4	4	8
b^4	5	5	10
x	1	2	3
y	3	3	8
z	1	5	8
t	1	3	10
u	1	2	1

Table 1. Performance of limit profiles b^1, b^2, b^3, b^4 and alternatives x, y, z, t, u

	b^1	b^2	b^3	b^4	Assignment
x	\succeq	$\not\succeq$	$\not\succeq$	$\not\succeq$	C_1
y	\succeq	\succeq	$\not\succeq$	$\not\succeq$	C_2
z	\succeq	\succeq	\succeq	$\not\succeq$	C_3
t	\succeq	\succeq	$\not\succeq$	$\not\succeq$	C_2
u	\succeq	$\not\succeq$	$\not\succeq$	$\not\succeq$	C_1

Table 2. Comparison between alternatives and profiles and final assignments.

Note that the MR-sort method allows the use of heterogeneous scales since the only information being useful for the relation \succeq is an ordinal one. Hence, a problem with only ordinal scales or with different types of scales (interval one, ratio one and ordinal one) can be handled without difficulty by an MR-sort model.

3 Learning a MR-sort model from assignments

We assume that we are given some assignments of alternatives to categories. In this work we assume that limit profiles are given. In the following, we first introduce the notion of possible and necessary categories, and then we present a maximum-margin optimization for learning the parameters of a MR-sort model.

3.1 Possible Categories

Let \mathcal{A} be the set of alternatives whose category is known and LS the “learning set” (pairs of alternatives and categories): $LS = \{(x, C_h), x \in \mathcal{A}, x \rightarrow C_h\}$. From LS, assuming an underlying MR-sort model, we can reason about the weights that are consistent with the current knowledge. According to Equation (1) and (2), an assignment of the type $x \xrightarrow{DM} C_h$ (made by the decision maker) corresponds to imposing the linear

constraints:

$$\sum_{i: x_i \geq b_i^h} w_i \geq \lambda \quad (3)$$

$$\sum_{i: x_i \geq b_i^{h+1}} w_i < \lambda \quad (4)$$

Let $\Theta(\text{LS})$ be set of parameters that are compatible with the learning set LS, satisfying constraints (3) and (4) for all pairs in LS, and the requirements that the weights are non negative and normalized.

When dealing with partially specified preference models, it is typical to reason about possible and necessary preference information [13]. In our context, given an alternative and its performance evaluation we can reason about the parameters that are consistent with the current knowledge.

Now, given an item $x \notin \mathcal{A}$, we define the set of possible categories $\text{PC}(x; \text{LS})$ as the set of categories such that there is an instantiation of the parameters $\theta = \{w_1, \dots, w_n, \lambda\}$ consistent with the assignment *given the previously known assignments* LS.

$$\text{PC}(x; \text{LS}) = \{C_i \in \mathcal{C} \mid \exists \theta \in \Theta(\text{LS}) : x \xrightarrow{\theta} C_i\}$$

where we write $x \xrightarrow{\theta} C_i$ to emphasize the dependency between the parameters θ and the assignment. If there is only one possible category, i.e. $|\text{PC}(x)| = 1$, then it means that the alternative has to be *necessarily* assigned to the only category in PC (assuming that the model is consistent with the learning set; in Section 3.2 we discuss how to handle inconsistencies). In practice it is possible that a partial knowledge about the parameters allows us to determine in which categories we have to place several alternatives. We anticipate that the concept of possible categories plays an important role in our elicitation strategies (see Section 4.2).

3.2 Maximum margin optimization

We now address the problem of learning the parameters $\theta = \{w_1, \dots, w_n, \lambda\}$ of a MR-sort model (given a set of learning assignments) with a linear program. We assume that we are given as input the following values: the number of criteria n , the number of categories p , and the limit profiles b^h for each $h \in [p]$. Our goal is to assess the weights w and the majority threshold λ , that are decision variables for the optimization problem. In order to discriminate between different choices for the parameters, the first step is to introduce a shared margin μ as an additional decision variable. The first linear program

(LP) assumes that the data in LS is perfectly consistent with the MR-sort model.

$$\mu^* = \max \quad \mu \quad (5)$$

$$\text{s.t.} \quad \sum_{i:x_i \geq b_i^h} w_i \geq \lambda + \mu \quad \forall (x, C_h) \in \text{LS} \quad (6)$$

$$\sum_{i:x_i \geq b_i^{h+1}} w_i \leq \lambda - \mu \quad \forall (x, C_h) \in \text{LS} \quad (7)$$

$$\sum_{i=1}^n w_i = 1 \quad (8)$$

$$\mu \in \mathbb{R} \quad (9)$$

$$w_i \geq 0 \quad \forall i \in [n] \quad (10)$$

Constraints (6) and (7) correspond to Equations (3) and (4) with the inclusion of the shared margin μ . Note that later in Section 4.2 we will use $\mu^*(\text{LS})$ to denote the application of the LP above to find the value of the best compatible margin.

We observe that in general there may be occasional inconsistencies in user feedback, or simply the data may not be compatible with a MR-sort model. The “typical” way to handle this is to introduce slack variables (whose sum we aim at minimizing); this idea goes back to the UTA approach [14] and several later models [12,1]. We formalize the problem of finding the weights of a MR-sort model (given a set of learning assignments) with the following linear program.

$$m^* = \max \quad \mu - \alpha \sum_{(x, C_h) \in \text{LS}} \xi_{i,j} \quad (11)$$

$$\text{s.t.} \quad \sum_{i:x_i \geq b_i^h} w_i + \xi_{x,h} \geq \lambda + \mu \quad \forall (x, C_h) \in \text{LS} \quad (12)$$

$$\sum_{i:x_i \geq b_i^{h+1}} w_i - \xi_{x,h} \leq \lambda - \mu \quad \forall (x, C_h) \in \text{LS} \quad (13)$$

$$\sum_{i=1}^n w_i = 1 \quad (14)$$

$$\mu \in \mathbb{R} \quad (15)$$

$$\xi_{x,h} \geq 0 \quad \forall (x, C_h) \in \text{LS} \quad (16)$$

$$w_i \geq 0 \quad \forall i \in [n] \quad (17)$$

The variables are the following: $\lambda \in [0, 1]$, $w_i \in [0, 1] \forall i \in [n]$, $\mu \in [0, 1]$ and $\xi_{x,h} \in [0, 1] \forall x \in \text{LS}$ such that x assigned to C_h . We use a parameter α to express the “cost” of violating a constraint representing assignments from the learning set. The objective (Equation 11) is to maximize the shared margin μ (that we want to maximize) minus the sum of the constraints violations $\xi_{i,j}$ (that we want to minimize). In order to be able to handle the inconsistency in the user’s answers we add a slack variable

$(\xi_{x,h})$ for each $x \in \mathcal{A}$ and for each $h \in \{1, \dots, p\}$. Constraint 12 and constraint 13 handle the conditions related to the assignments of the alternatives in the learning set LS. Constraint 14 enforces that the weights are normalized.

4 Incremental MR-sort

Algorithm 1 The MR-sort-Inc algorithm. K is the number of questions, X the set of alternatives, C the set of categories, LS ($LS = \{(x, C_h), x \in \mathcal{A}, x \rightarrow C_h\}$) learning assignments, s the strategy.

```

1: Procedure MR-sort-Inc( $K, X, C, LS, s$ )
2: for  $j \leftarrow 1, \dots, K$  do
3:   Compute  $m^*(LS)$  and associated  $w^*$  and  $\lambda^*$  with linear program
4:   Classify items in  $X \setminus \mathcal{A}$  using  $w^*$  and  $\lambda^*$ 
5:   Compute classification error ▷ This is only possible in simulations
6:   for  $x \in X \setminus \mathcal{A}$  do ▷ Evaluate items to decide next question
7:     for  $C_i \in C$  do
8:        $LS' \leftarrow LS \cup \{x \rightarrow C_i\}$  ▷ Include additional assignment
9:        $v_i(x) \leftarrow m^*(LS')$  ▷ Compute margin a posteriori
10:    end for
11:     $S_s(x) \leftarrow$  aggregate  $v(x)$  according to strategy  $s$ 
12:  end for
13:   $x^* \leftarrow \arg \max_x (S_s(x))$  ▷ Decide what to ask next
14:   $C_{x^*} \leftarrow \text{Answer}(x^* \rightarrow ?)$  ▷ Ask question to DM
15:   $LS \leftarrow LS \cup \{(x^*, C_{x^*})\}$ 
16: end for
17: return  $w^*, \lambda^*$ 

```

In this section, we provide an elicitation method for an ordinal classification problem assuming that the preferences can be modeled by MR-sort.

4.1 Main Framework

Our elicitation procedure starts with a small learning set LS to which we add one-by-one new assignment examples. We ask questions of the type “*In which category should x be assigned to?*”; it is crucial to select “informative” items to ask about, in order to quickly converge (in few interaction cycles) to a good classification model.

We remind that a MR-sort model is defined by the following parameters: the weights (w_1, \dots, w_n) , the limit profiles (b^1, \dots, b^{p+1}) and majority threshold λ . We fix the limit profiles before the beginning of the elicitation. In the experiments below the limit profiles are chosen in a way to evenly partition the criteria scale. The main steps of our approach are presented in Algorithm 1. Briefly, we start with some data from the

learning set LS (learning alternatives \mathcal{A} and their classification) and we include them as constraints in our linear program (LP). Given the learning set, we can estimate the weights w^* and λ^* using the linear program; in simulations we can also evaluate the classification error. After that, we test the assignment of each unassigned item to each category and retrieve the margin values obtained when using the optimization routine with each of the additional assignments. This gives us a vector of margins “a posteriori”, that are aggregated differently depending on the “strategy” (different strategies will be discussed in the following section), giving a score for each item. The item having the highest score is chosen: the user is asked about the assignment of that item and the learning set is augmented. This procedure continues until a stopping condition (in our experiments when we reach a fixed number of questions, but in real applications termination may be left to the user). The last linear program contains all the assignments of the computed learning set, hence it provides the weights that we are looking for.

Example. We apply our incremental elicitation algorithm to the running example. Suppose that we only know the assignment of alternative x ($LS = \{(x, C_1)\}$) and we want to ask just one question to the user. We first set the limit profiles: $b^1 = (0, 0, 0)$, $b^2 = (\frac{5}{3}, \frac{5}{3}, \frac{10}{3})$, $b^3 = (\frac{10}{3}, \frac{10}{3}, \frac{20}{3})$, $b^4 = (5, 5, 10)$. Our goal is to learn λ and a weight vector w' which will assign the remaining alternatives to categories as close as possible to the ones presented in Example 2. For this, according to our algorithm, we compute the score of alternatives y, z, t, u ($S_s(y), S_s(z), S_s(t), S_s(u)$) and ask the assignment of the alternative having the highest score. We add this new constraint to LS and find the weight vector w^* corresponding to the largest margin. Using w^* , we can find the current best assignment of the remaining alternatives and compute ⁴ an error measure using the assignments of Example 1 and the one that we find using w^* .

4.2 Question selection strategies

In the following we present several strategies to select the next question to ask to the user. These strategies are used within our interactive elicitation paradigm described in Algorithm 1. The proposed strategies makes use of the max-margin optimization programs (discussed above in Subsection 3.2) to identify informative questions.

Most Uncertain With this strategy, we aim to ask the DM to classify the *most uncertain* item in $X \setminus \mathcal{A}$, that is the item that is compatible (according to the constraints derived from the known assignments of the learning set) with the highest number of categories. The score that this strategy assigns to an alternative x is given by the cardinality of the set of possible categories: $S_{MU}(x) = |PC(x; LS)|$.

The computation of the set $PC(x)$ for an item x makes use of the linear program for learning the parameters of a MR-sort without the slack variables (Equations 5-10). For each category C_i , we then constrain the item x to be assigned to C_i and simply check whether the margin is non negative; indeed $\mu^*(LS \cup (x, C_i)) > 0$ if and only if C_i is a possible category ($C_i \in PC(x; LS)$). The score of this strategy can be compactly

⁴ This step is of course only to be performed in simulations, in real use of the procedure the classification error will not be known

written as:

$$S_{\text{MU}}(x) = |\text{PC}(x; \text{LS})| = \sum_{i=1}^p H[\mu^*(\text{LS} \cup (x, C_i))]$$

where $H(\cdot)$ is the step function.

Example. Consider the problem presented in Example 2. We suppose that we know the performance of each alternative, and the assignment of x , while the other assignments are unknown. Assume that we want to ask just one question. As before, we fix $b^1 = (0, 0, 0)$, $b^2 = (\frac{5}{3}, \frac{5}{3}, \frac{10}{3})$, $b^3 = (\frac{10}{3}, \frac{10}{3}, \frac{20}{3})$, $b^4 = (5, 5, 10)$. Table 3 presents the S_{MU} of all the remaining alternatives (step 3).

Alternatives	PC(x ; LS)	S_{MU}
y	$\{C_2, C_3\}$	2
z	$\{C_1, C_3\}$	2
t	$\{C_1, C_2, C_3\}$	3
u	$\{C_1\}$	1

Table 3. Number of possible categories for the example of hotel categorization.

As a result we ask the assignment of alternative t to the user. The user will answer C_2 to this question (coherent with Example 2) and the constraint related to $t \rightarrow C_2$ will be added to the LP. After the inclusion of the constraint $t \rightarrow C_2$ in LS, we find our final weight vector w^* , such as $w^* = (0.266, 0.366, 0.366)$. Even if w' is different from the weight vector of Example 2 ($w = (0.2, 0.5, 0.3)$), after our incremental elicitation we find the same assignments for y, z, u (i.e. $y \rightarrow C_2, z \rightarrow C_3$ and $u \rightarrow C_1$).

Sum-of-margin strategy The problem of the most-uncertain strategy is that, roughly speaking, it is agnostic to whether a potential assignment is consistent with large portions of the parameter space or just with small area. The intuition of sum-of-margin is to use the value of the objective function of the LP as a surrogate measure of the “degree” of satisfaction of an assignment $x \rightarrow C_h$. Intuitively, we should ask about items that may fit well into several different categories. We also include the possibility that some constraints can be violated, therefore make use of the second LP (Equations 11-17), with the penalty variables ξ for violated assignments of the learning set.

Considering an alternative $x \in X \setminus \mathcal{A}$, we evaluate the penalized margin m^* adding the assignment $x \rightarrow C_i$ for all $i \in [p]$ and construct the following vector:

$$v(x) = (m^*(\text{LS} \cup (x, C_1)), \dots, m^*(\text{LS} \cup (x, C_p))).$$

In order to aggregate this vector into a single numerical measure, we adopt the sum. Hence, the score of the alternative x is computed as

$$S_{\Sigma}(x) = \sum_{i=1}^p v_i(x) = \sum_{i=1}^p m^*(\text{LS} \cup (x, C_i)).$$

Assignment	$v_i(\cdot)$	S_Σ	Assignment	v'	S_E
$y \rightarrow C_1$	-0.45	} 0.45	$y \rightarrow C_1$	0	} 0.30
$y \rightarrow C_2$	0.45		$y \rightarrow C_2$	0.5	
$y \rightarrow C_3$	0.45		$y \rightarrow C_3$	0.5	
$z \rightarrow C_1$	0.45	} 0.90	$z \rightarrow C_1$	0.5	} 0.30
$z \rightarrow C_2$	0		$z \rightarrow C_2$	0	
$z \rightarrow C_3$	0.45		$z \rightarrow C_3$	0.5	
$t \rightarrow C_1$	0.45	} 1.08	$t \rightarrow C_1$	0.43	} 0.44
$t \rightarrow C_2$	0.18		$t \rightarrow C_2$	0.15	
$t \rightarrow C_3$	0.45		$t \rightarrow C_3$	0.42	
$u \rightarrow C_1$	0.49	} -0.10	$u \rightarrow C_1$	1	} 0.00
$u \rightarrow C_2$	0		$u \rightarrow C_2$	0	
$u \rightarrow C_3$	-0.55		$u \rightarrow C_3$	0	

Table 4. The score obtained by the different items in the hotel categorization example using the heuristics sum-of-maring (left) and entropy (right)

Example. Consider again the running example. Table 4 (in the left) presents the S_Σ of alternatives $\{y, z, t, u\}$. As the example shows, the best question is to ask about t , the second best question is to ask about z , then y , and finally u . We note a disagreement of MU and sum-of-margin about the ranking of z and y .

Entropy This strategy adopts the notion of entropy to assess the uncertainty for a given alternative. As the strategy sum-of-margin, we calculate m^* which represents the maximum penalized margin if we assign the alternative x to the category C_i , for all $i \in [p]$. We then combine these values in an evaluation vector v' that filters out negative values (assigning them a zero value) and we normalize so that the values sum up to one:

$$v'_i(x) = \frac{R(m^*(\text{LS} \cup (x, C_i)))}{\sum_{j=1}^p R(m^*(\text{LS} \cup (x, C_j)))}$$

where $R(\cdot)$ is the ramp function. We then apply the entropy to calculate the final score of the alternative.

$$S_E(x) = - \sum_{i \in [p]: v'_i(x) > 0} v'_i(x) \log(v'_i(x))$$

Intuitively this method should favour asking about items whose uncertainty is the greatest in an information-theoretic sense, considering the vector v' (composed of normalized non-negative margin values) as a surrogate measure for the probability distribution of the category of x (i.e. how likely is x to be in each of the categories).

Example. We consider again our example on hotel categorization. We now apply the entropy strategy; Table 4 (on the right) presents the score S_E of all the alternatives. Item t has the highest score, then z and y are tied in the second position; u is last.

Random strategy As a baseline, we consider the random selection of an object x from $X \setminus \mathcal{A}$. Utilizing this strategy is equivalent to use a non incremental elicitation procedure with $|LS| + K$ (LS being the initial learning set of Algorithm 1 and K being the number of questions) alternatives in the learning set.

Note that all our strategies have the same computational complexity : $O(Kmp)$ (resp. number of questions, alternatives, categories).

5 Experiments

We implemented our incremental MR-sort algorithm using Java API of CPLEX for solving the LP.⁵ We performed simulations aimed at evaluating the effectiveness of the proposed elicitation strategies with two data-sets: a synthetic data-set fully represented by a MR-sort model, and a data-set from the UCI machine learning repository [8].

Synthetic data-set For the first part of our experiments, the input data was generated randomly using a uniform distribution on the space of evaluations.

We include only one assignment, chosen at random in LS. The steps of the simulation are as follows:

- i. Using a uniform distribution for each of the parameters, we generate the performance table of alternatives, a vector of weight w , the majority threshold λ . The limit profiles are chosen to evenly partition the range of evaluation values. These parameters fully specify the decision maker; we call this model \bar{M} . We then apply the MR-sort method in order to find the assignments of the generated alternatives; these assignments constitute the “ground truth”.
- ii. We apply our incremental elicitation Algorithm 1 using the assignments of some alternatives of \bar{M} as learning assignments. We apply incremental MR-sort by asking questions to the user based on the proposed strategies, simulating a decision maker who answers according to the results of the \bar{M} . We generate as output a vector of weight w^* representing the DM’s preferences.
- iii. Using the assumed limit profiles and the learned weight vector w^* and the majority threshold λ^* , we obtain our learned model M that provides the assignments of the remaining alternatives.
- iv. At the end we calculate an error rate based on the difference between the assignments of the true model \bar{M} and the learned model M .

We iterate these steps 50 times and we evaluate the classifications obtained with the different methods according to the average classification error $AE_k = \sum_{x \in X} \text{err}_k(x)/m$ where m is the number of alternatives and $k \in \{1, 2\}$. The value $\text{err}_k(x) = d(C_i, C_j)$ is the magnitude of error when $x \xrightarrow{\bar{M}} C_i$ and $x \xrightarrow{M} C_j$ (the true model assigns x to C_i while the learned model M assigns it to C_j). err_1 adopts a 0/1 loss as distance, while err_2 considers the displacement between the assignments, that is $d(C_i, C_j) = |i - j|$.

Figure 1 presents the results of our simulations (for more results see figures 4 - 6 in the appendix). Note that the random strategy performs very poorly, while all three

⁵ All experiments were run on a 2.9 GHz Intel, Core i7 and 16 Giga of RAM.

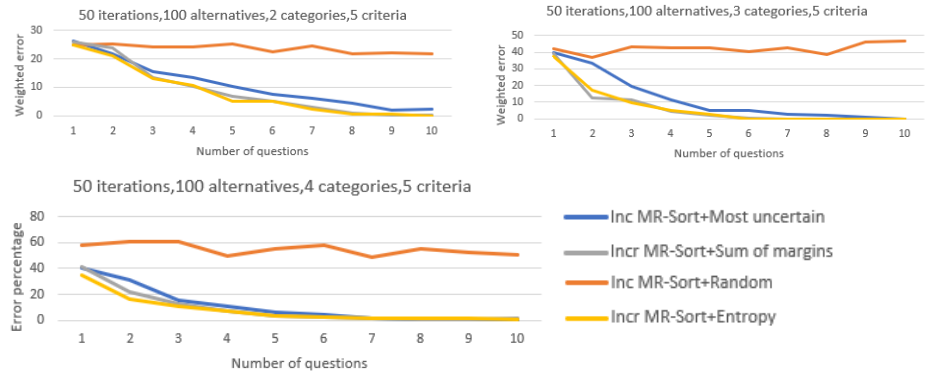


Fig. 1. AE_2 based on number of questions with $m = 100$, $n = 5$, $p = 2$, $p = 3$, $p = 4$, $|LS| = 1$ and 50 iterations

strategies (maximum uncertainty, sum of margin, entropy) have reasonably good performance. Not surprisingly, the higher the number of categories, the quicker we converge, since there is more uncertainty with more categories. Conversely, the higher the number of criteria the less quickly the interaction converges. The results with respect to AE_2 are quite satisfying since they show that, after few questions have been answered, our procedure makes few assignment mistakes and these mistakes concern consecutive categories most of the times.

UCI Car We performed simulations based on Car data-set. There are 1728 items defined on 6 categorical attributes (that we interpret as ordinal); the items are partitioned in 4 categories: unacceptable (65 items), acceptable (1210), good (384) and very good (69). We converted the qualitative (ordinal) attributes to numeric attributes and we set the criteria rating scale between 1 and 4. The steps of the simulation are : i.) We first compute the best error rate that we could have if all the data was included in LS . ii.) We ask questions until we reach this error percentage. We started to test on 300 alternatives chosen randomly and we start by including in the learning set only one assignment taken randomly.

In order to find the best error percentage, we call it P^* , we include all the data in LS and we vary the value of λ . We obtain P^* when λ is between 0.7 and 0.75. In this case the $err_1 = 18.86\%$ (err_2 being 0.24). So we aim to reach this percentage by asking the minimum number of questions. Our simulations showed that fixing the value of λ makes faster the convergence, we show in Figures 2- 3 the comparison between the 4 strategies where λ is fixed to 0.7.

With the random strategy we observe that the variation of error according to number of questions is not monotonous and we don't approach to P^* by asking 10 questions. On the other hand, the strategies maximum uncertainty and sum of margin are closer to P^* starting from 6th question. The strategy maximum uncertainty stops after the 7th question because it hangs in case of inconsistency. We notice that the assignments errors are almost always between consecutive categories; Figure 3 shows the performance with respect to the second metric.

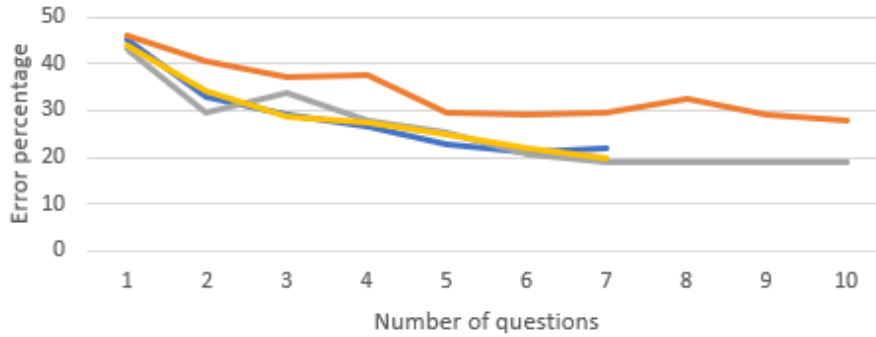


Fig. 2. AE_1 based on number of questions with $m = 300$, $n = 6$, $p = 4$, $|LS|=1$ and 50 iterations

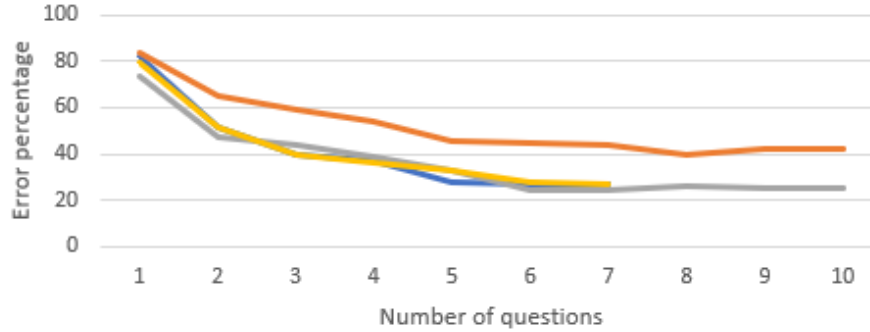


Fig. 3. AE_2 based on number of questions with $m = 300$, $n = 6$, $p = 4$, $|LS|=1$ and 50 iterations

6 Conclusions

MR-sort is an attractive method for ordinal classification that offers the advantage of allowing the use of heterogeneous scales (mixing ordinal and cardinal scales). In this paper we presented an incremental elicitation procedure for the parameters (weights and threshold) of MR-sort. Our approach relies on a maximum-margin optimization that aims at satisfy as well as possible the known assignments, following approaches proposed in the literature for non-interactive learning of MR-sort and variations [15,21,22]. The novelty of this paper consists in showing how the optimization can be used to evaluate the uncertainty associated to the items whose category is not known yet; the choice of the next question is based on evaluating the margin a posteriori (the value of maximum-margin optimization when adding a possible response to a question). Based on this intuition we proposed several strategies for selecting the next question to ask. We evaluated the proposed interactive elicitation procedure comparing the performance of the different strategies showing that the procedure quickly converges to the real optimal classification in very few interaction cycles.

We highlight that one important limitation of our framework is that we expect the decision maker to provide us with the limit profiles of the different categories. In future works we will relax this assumption considering techniques to elicit as well the limit profiles (either in a preliminary step or in an integrated approach), therefore providing a complete method for interactive elicitation of a MR-sort model. This task will be challenging, since previous works on (non incremental) elicitation of MR-sort have shown that, while it is possible to introduce integer variables [15] the resulting optimization is computationally very demanding and not scalable; therefore randomized heuristics [20,23] have been proposed.

We are also interested in performing simulation to compare our approach to other elicitation frameworks, as the recent work of Olteanu [18], and the approaches based on minimax regret [2,3], in realistic settings. We plan to investigate the connection between our approach and the field of machine learning. First, maximum-margin bears much similarity to Support Vector Machines (SVM). Second, ordinal classification has also been studied in machine learning. Third, there is strong similarity between incremental elicitation and active learning.

Finally, another important direction is to elicit the preferences of several users, providing methods that can exploit the similarity between users, as in Teso et al. [26].

References

1. Julien Ah-Pine, Brice Mayag, and Antoine Rolland. Identification of a 2-additive bi-capacity by using mathematical programming. In *Proceedings of ADT*, pages 15–29, 2013.
2. Nawal Benabbou, Patrice Perny, and Paolo Viappiani. A regret-based preference elicitation approach for sorting with multicriteria reference profiles. In *From Multicriteria Decision Making to Preference Learning (DA2PL'16)*, 2016.
3. Nawal Benabbou, Patrice Perny, and Paolo Viappiani. Incremental elicitation of choquet capacities for multicriteria choice, ranking and sorting problems. *Artif. Intell.*, 246:152–180, 2017.
4. Craig Boutilier. Computational decision support: Regret-based models for optimization and preference elicitation. *Crowley P. H. Zentall T. R. (Eds.), Comparative Decision Making: Analysis and Support Across Disciplines and Applications*, pages 423–453, 2013.
5. Denis Bouyssou and Thierry Marchant. An axiomatic approach to noncompensatory sorting methods in mcdm, ii: More than two categories. *European Journal of Operational Research*, 178(1):246 – 276, 2007.
6. Vincent Conitzer and Tuomas Sandholm. Vote elicitation: Complexity and strategy-proofness. In *AAAI/IAAI*, pages 392–397, 2002.
7. Carmel Domshlak, Eyke Hüllermeier, Souhila Kaci, and Henri Prade. Preferences in AI: An overview, 2011.
8. Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.
9. Eibe Frank and Mark A. Hall. A simple approach to ordinal classification. In *Proceedings of ECML*, pages 145–156, 2001.
10. Johannes Fürnkranz and Eyke Hüllermeier. *Preference Learning*. Springer-Verlag, Berlin, Heidelberg, 1st edition, 2010.
11. Johannes Fürnkranz, Eyke Hüllermeier, Eneldo Loza Mencía, and Klaus Brinker. Multilabel classification via calibrated label ranking. *Machine learning*, 73(2):133–153, 2008.
12. Krzysztof Gajos and Daniel S. Weld. Preference elicitation for interface optimization. In *Proceedings of UIST*, pages 173–182. ACM, 2005.

13. Salvatore Greco, Vincent Mousseau, and Roman Slowinski. Ordinal regression revisited: Multiple criteria ranking using a set of additive value functions. *European Journal of Operational Research*, 191(2):416–436, 2008.
14. E. Jacquet-Lagrèze and Y. Siskos. Assessing a set of additive utility functions for multicriteria decision making: the UTA method. *European Journal of Operational Research*, 10:151–164, 1982.
15. Agnès Leroy, Vincent Mousseau, and Marc Pirlot. Learning the parameters of a multiple criteria sorting method. In *Algorithmic Decision Theory - Second International Conference, ADT 2011, Piscataway, NJ, USA, October 26-28, 2011. Proceedings*, pages 219–233, 2011.
16. Vincent Mousseau, Roman Slowinski, and Piotr Zielniewicz. A user-oriented implementation of the ELECTRE-TRI method integrating preference elicitation support. *Computers & OR*, 27(7-8):757–777, 2000.
17. Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.
18. Alexandru Liviu Olteanu. Strategies for the incremental inference of majority-rule sorting models. In *DA2PL'2018: from Multiple Criteria Decision Aid to Preference Learning*, Poznan, Poland, November 2018.
19. Bernard Roy. The outranking approach and the foundations of electre methods. In *Readings in multiple criteria decision aid*, pages 155–183. Springer, 1990.
20. O. Sobrie, V. Mousseau, and M. Pirlot. Learning a majority rule model from large sets of assignment examples. In P. Perny, M. Pirlot, and A. Tsoukiás, editors, *Algorithmic Decision Theory*, Lecture Notes in Artificial Intelligence, pages 336–350. Springer, 2013.
21. Olivier Sobrie, Vincent Mousseau, and Marc Pirlot. Learning the parameters of a non compensatory sorting model. In *International Conference on Algorithmic Decision Theory*, pages 153–170. Springer, 2015.
22. Olivier Sobrie, Vincent Mousseau, and Marc Pirlot. A population-based algorithm for learning a majority rule sorting model with coalitional veto. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 575–589. Springer, 2017.
23. Olivier Sobrie, Vincent Mousseau, and Marc Pirlot. Learning monotone preferences using a majority rule sorting model. *International Transactions in Operational Research*, 26(5):1786–1809, 2019.
24. Hossein Azari Soufiani, David C Parkes, and Lirong Xia. Preference elicitation for general random utility models. *arXiv preprint arXiv:1309.6864*, 2013.
25. Stefano Teso, Andrea Passerini, and Paolo Viappiani. Constructive preference elicitation by setwise max-margin learning. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 2067–2073, 2016.
26. Stefano Teso, Andrea Passerini, and Paolo Viappiani. Constructive preference elicitation for multiple users with setwise max-margin. In *International Conference on Algorithmic Decision Theory*, pages 3–17. Springer, 2017.
27. Zhibing Zhao, Haoming Li, Junming Wang, Jeffrey Kephart, Nicholas Mattei, Hui Su, and Lirong Xia. A cost-effective framework for preference elicitation and aggregation. *arXiv preprint arXiv:1805.05287*, 2018.

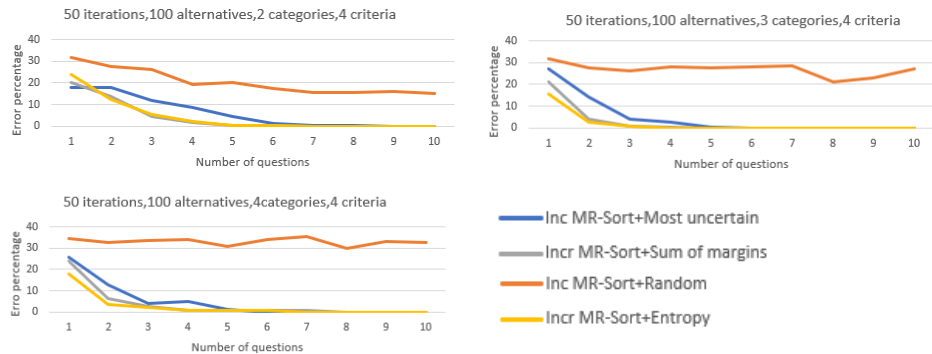


Fig. 4. Classification error AE_1 based on number of questions with $m = 100, n = 4, p = 2, p = 3, p = 4, |LS| = 1$ and 50 iterations

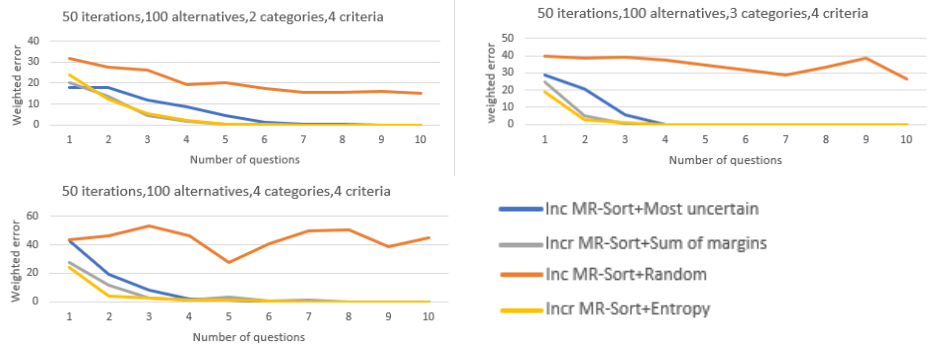


Fig. 5. AE_2 based on number of questions with $m = 100, n = 4, p = 2, p = 3, p = 4, |LS| = 1$ and 50 iterations

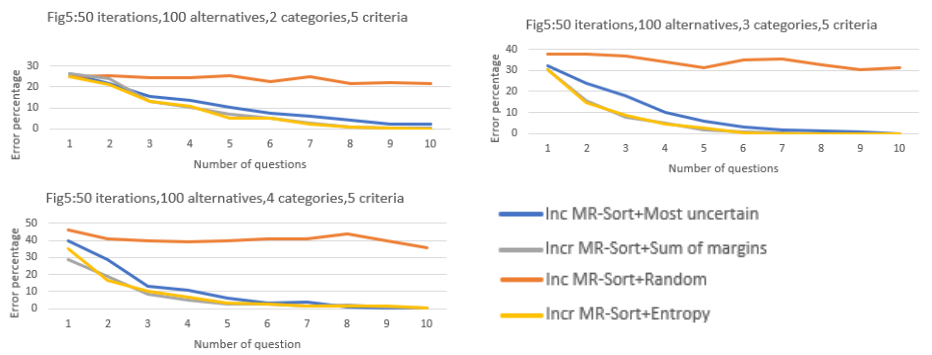


Fig. 6. AE_1 based on number of questions with $m = 100, n = 5, p = 2, p = 3, p = 4, |LS| = 1$ and 50 iterations