Learning Preference Models with Sparse Interactions of Criteria

Margot Herin\textsuperscript{1}, Patrice Perny \textsuperscript{1}, Nataliya Sokolovska\textsuperscript{2}
\textsuperscript{1}Sorbonne University, CNRS, LIP6, Paris, France
\textsuperscript{2}Sorbonne University, CNRS, LCQB, Paris, France
\{margot.herin,patrice.perny\}@lip6.fr, nataliya.sokolovska@sorbonne-universite.fr

Abstract
Multicriteria decision making requires defining the result of conflicting and possibly interacting criteria. Allowing criteria interactions in a decision model increases the complexity of the preference learning task due to the combinatorial nature of the possible interactions. In this paper, we propose an approach to learn a decision model in which the interaction pattern is revealed from preference data and kept as simple as possible. We consider weighted aggregation functions like multilinear utilities or Choquet integrals, admitting representations including non-linear terms measuring the joint benefit or penalty attached to some combinations of criteria. The weighting coefficients known as Möbius masses model positive or negative synergies among criteria. We propose an approach to learn the Möbius masses, based on iterative reweighted least square for sparse recovery, and dualization to improve scalability. This approach is applied to learn sparse representations of the multilinear utility model and conjunctive/disjunctive forms of the discrete Choquet integral from preferences examples, in aggregation problems possibly involving more than 20 criteria.

1 Introduction
One of the main challenges of preference modeling in the context of multicriteria decision making is to construct simple and explainable decision models keeping sufficient flexibility to accurately model human preferences and decision behaviors. In the field of multiattribute/multicriteria decision making, the presence of possible interactions among criteria is a source of complexity for preference modeling because it prevents representing preferences by simple linear models such as weighted arithmetic means. More sophisticated weighted evaluation models including non-linear terms measuring the joint benefit or penalty attached to some groups of criteria are needed. Interactions may be represented by product terms as in the multilinear utility model [Keeney et al., 1993], or by minimum or maximum operations as in the Choquet integral [Grabisch et al., 2009], or possibly by other monotonic non-linear factors involving several attributes or criteria.

For example, if the decision maker (DM) prefers solutions with balanced utility vectors then one cannot simply use a weighted arithmetic mean of type \( \sum_{i=1}^{n} w_i x_i \) to define the overall utility attached to a utility vector \((x_1, \ldots, x_n)\). For instance, it would not be possible to make \((0.5, 0.5)\) better than \((1, 0)\) and \((0, 1)\) simultaneously. A simple solution in this case may be to allocate a bonus to alternatives that perform well on all criteria. It is sufficient to add a conjunctive term to the weighted mean, e.g., \( \prod_{i=1}^{n} x_i \) or \( \min\{x_1, \ldots, x_n\} \), weighted by a sufficiently large positive coefficient. We observe indeed that this conjunctive term takes a strictly positive value on \((0.5, 0.5)\) but remains null on \((1,0)\) and \((0,1)\). More generally, beyond the linear part of the aggregation function, various interaction terms could be inserted to model positive but also negative synergies within some groups of criteria.

However, allowing the possibility of interactions in a decision model is a source of complexity in preference modeling and preference learning due to the combinatorial nature of these interactions. In an aggregation model involving \( n \) criteria, interactions may appear in any of the \( 2^n - n - 1 \) subsets of criteria including more than one element. For \( n = 10 \) criteria it represents slightly more than 1000 possible interactions to analyze. When \( n = 20 \) it already represents more than one million possible interactions. In order to preserve scalability in learning the interactions, a standard approach is to reduce the combinatorial aspect of the problem by allowing only a limited number of them. For example one can just consider pairwise interactions of criteria. More generally one could limit interactions to subsets of size \( k \) for some \( k \) significantly smaller than \( n \). However, this prior restriction eliminates very simple and natural representations of preferences that require larger interactions. For example, in the above example, we have shown that the introduction of a conjunctive term including all criteria may be natural to promote balanced solutions. Besides, another simple example is the Hurwicz criterion [Hurwicz, 1951] which is standardly used to make a tradeoff between the worst and the best components. It is defined as a convex combination of \( \min\{x_1, \ldots, x_n\} \) and \( \max\{x_1, \ldots, x_n\} \) and therefore includes two interactions terms. Both of them involve the entire set of criteria and cannot be simply approximated by interactions on smaller sets.
Here we would like to propose another approach where no prior restriction on the possible interaction groups is made. The useful groups will emerge from preference data with the aim of constructing a model as simple as possible, that fits well the preference examples. In this perspective, we consider an aggregation function defined as a weighted sum of factors made of criteria in interaction and propose a method to learn a sparse representation of these weights.

The paper is organized as follows: In Section 2 we recall some background on multicriteria aggregation models including interacting components. Then in Section 3 we introduce an approach to obtain sparse representations of interactions by iterative reweighted least square regularization, and dualization to improve the scalability of the approach. Section 4 presents some numerical tests to evaluate the performance of the proposed approach both in terms of computation time and generalizing performances.

2 Evaluation Models with Interacting Criteria

The general framework. In a multidimensional decision problem, the alternatives are described with respect to \( n \) points of views \((n > 1)\) that must be considered in the evaluation process. Depending on the decision context, these viewpoints may refer to different attributes describing the alternatives (multitribute decision making), different criteria used to compare the alternatives (multicriteria decision making), or different individuals expressing their preferences (multigen agent decision making). In any case, every alternative \( x \) is described by a vector \((c_1(x), \ldots, c_n(x))\) of consequences where \(c_i(x)\) represents the value of \( x \) with respect to the \( i \)th viewpoint. Let \( N = \{1, \ldots, n\} \) denote the index set of viewpoints. Let \( X_i \) denote the set of possible consequences on the \( i \)th viewpoint for all \( i \in N \) and \( X = X_1 \times \ldots \times X_n \) the set of all possible consequence vectors. Let \( \succsim \) be the preference relation of the DM over \( X \). One standard approach in preference modeling consists of representing \( \succsim \) by a decomposable function \( u \) on \( X \) of the form:

\[
u(x) = F(u_1(c_1(x)), \ldots, u_n(c_n(x)))\]

where \( u_i : X_i \rightarrow [0, 1], i \in N \) are marginal utility functions representing the attractiveness of consequences \( c_i(x) \) for the DM and \( F : [0, 1]^n \rightarrow [0, 1] \) is an aggregation function non-decreasing in each argument. Function \( u \) is said to represent \( \succsim \) when \( x \succsim y \) if and only if \( u(x) \geq u(y) \). Throughout the paper, \( \succ \) is used to denote the asymmetric part (strict preference) of \( \succsim \), whereas \( \sim \) denotes its symmetric part (indifference).

Let us recall two standard examples of function \( u \), widely used to represent preferences in multitribute/multicriteria decision problems involving criteria in interaction:

Example 1. The multitribute utility model defined by:

\[
ML_u(x) = \sum_{S \subseteq N} v(S) \prod_{i \in S} u_i(c_i(x)) \prod_{i \notin S} (1 - u_i(c_i(x)))
\]

was introduced in the context of multitribute decision making under risk [Keeney et al., 1993] but is also used and axiomatically justified in the context of multitribute decision making [Dyer and Sarin, 1979] and [Grabisch, 2016, Chap. 6]. In the context of this paper, \( v \) is a set function defined on the power set \( 2^N \) and valued in the unit interval, assigning a weight to any subset of viewpoints. Another well-known decision model defined from a set function \( v \) is the following:

Example 2. The discrete Choquet integral

\[
C_v(x) = \sum_{i=1}^n \left[ v(X_i) - v(X_{i+1}) \right] u_i(c_i(x))
\]

where \((. \) is any permutation of \( N \) such that \( u_i(c_i(x)) \leq u_{i+1}(c_{i+1}(x)) \) and \( X_i = \{(i), \ldots, (n)\}, i \in N \) with \( x(0) = 0 \) and \( X_{n+1} = \emptyset \). For instance, if \( n = 3 \) and \( x \) is such that \( u_2(c_2(x)) \leq u_1(c_1(x)) \leq u_3(c_3(x)) \), then \( C_v(x) = [v(1, 2, 3) - v(1, 3)]u_2(c_2(x)) + [v(1, 3) - v(3)]u_3(c_3(x)) + v(3)u_1(c_1(x)) \).

The Choquet integral was initially introduced in the context of decision making under uncertainty [Schmeidler, 1989] but is also widely used in the context of multitribute/multicriteria decision making [Grabisch, 1996; Grabisch and Labreuche, 2010].

In both models (multilin and Choquet), one can assume that \( v \) is normalized, i.e., it satisfies boundary conditions \( v(0) = 0 \) and \( v(N) = 1 \). It is also generally assumed that \( v \) is monotonic with respect to set inclusion (which guarantees the monotonicity of \( u \) w.r.t. weak Pareto dominance). More formally, if \( v(A) \leq v(B) \) for all subsets \( A, B \subseteq N \) such that \( A \subseteq B \) then \( u_i(c_i(x)) \geq u_i(c_i(y)) \) for all \( i \in N \) implies \( u(x) \geq u(y) \) for all pairs \((x, y)\) whether \( u \) is defined by (2) or (3). Such monotonic set functions named capacities are widely used in preference aggregation [Grabisch et al., 2003; Grabisch, 2016]. The capacity provides a non-necessarily additive weighting system (the weight of a set is not necessarily the sum of the weights of its elements) and super-additivity and sub-additivity are used to model positive and negative synergies between the components of the decision model.

The capacity is a preference parameter that must be elicited by questioning the DM or learned from preference examples. The other preference parameters used in these models are marginal utility functions \( u_i, i \in N \) that can be obtained before the identification of the capacity. In the case of the multilinear model, marginal utilities can be elicited using standard gamble queries under mutual utility independence, an axiom usually assumed to justify the multilinear model under uncertainty [Keeney et al., 1993]. They can alternatively be derived from comparisons of preference intensities under weak difference independence, an axiom usually assumed to justify the multilinear model in multicriteria/multitribute decision making [Grabisch, 2016, Chap. 6]. For the Choquet integral, the utility functions can be obtained using standard sequences of tradeoff queries [Wakker and Deneffe, 1996], or constructed with the Macbeth method [Grabisch and Labreuche, 2010] or learned from preference examples [Herin et al., 2022b]. We assume here that functions \( u_i \) have been elicited beforehand using one of the above mentioned methods and we focus on learning the capacity from preference examples. From now on, any alternative \( x \) is described by the utility vector \( x = (x_1, \ldots, x_n) \in [0, 1]^n \) where \( x_i = u_i(c_i(x)), i \in N \).
Representations based on Möbius masses. A useful alternative representation of any capacity \( v \) is given by its Möbius transform \( m_v \) defined as follows:

\[
m_v(S) = \sum_{T \subseteq S} (-1)^{|S| - |T|} v(T) \quad \text{with} \quad v(S) = \sum_{T \subseteq S} m_v(T)
\]

The values \( m_v(S) \) are called Möbius masses. We remark that we necessarily have \( \sum_{T \subseteq \emptyset} m_v(T) = 1 \), since \( v(\emptyset) = 1 \).

It is interesting to note that \( M_L v(x) \) can be directly defined from Möbius masses as follows [Owen, 1975]:

\[
M_L v(x) = \sum_{S \subseteq N} m_v(S) \prod_{i \in S} x_i
\]

Similarly, \( C_v(x) \) admits several reformulations from \( m_v \) [Chateauneuf and Jaffray, 1989; Grabisch et al., 2009]:

\[
C_v(x) = \sum_{S \subseteq N} m_v(S) \min_{i \in S} \{x_i\} \quad \text{conjunctive form} \quad (5)
\]

\[
C_v(x) = \sum_{S \subseteq N} m_v(S) \max_{i \in S} \{x_i\} \quad \text{disjunctive form} \quad (6)
\]

where \( \bar{v} \) is the conjugate of \( v \), i.e., the capacity defined by \( \bar{v}(S) = v(N) - v(N \setminus S) \) for all \( S \subseteq N \).

Example 3. Let \( N = \{1, 2, 3\} \) and \( v, \bar{v} \) defined on \( N \) by:

\[
|S| 1 2 3 1,2,3
\]

\[
v(S) 0.1 0.2 0.3 0.4 0.5 1.0
m_v(S) 0.1 0.2 0.3 0.0 0.0 0.4
\bar{v}(S) 0.5 0.6 0.7 0.8 0.9 1.0
m_{\bar{v}}(S) 0.5 0.6 0.7 -0.4 -0.4 -0.4
\]

Let \( x = (x_1, x_2, x_3) \) with \( x_2 \leq x_1 \leq x_3 \). Then we have:

\[
M_L v(x) = 0.1 x_1 + 0.2 x_2 + 0.3 x_3 + 0.4 x_1 x_2 x_3 \quad (Eq. 4)
\]

\[
C_v(x) = 0.1 x_1 + 0.2 x_2 + 0.3 x_3 + 0.4 x_2 \quad (Eq. 5).
\]

Here the disjunctive form of \( C_v \) (Eq. 6) is less interesting because \( m_{\bar{v}} \) is less sparse than \( m_v \). The converse holds for \( C_v \). We have \( C_v(x) = 0.1 x_1 + 0.2 x_2 + 0.3 x_3 + 0.4 x_3 \) by Eq. 6 whereas the conjunctive form based on \( m_v \) is less compact.

In order to factorize and generalize Equations 4-6, we will now consider a general decision model of the form:

\[
F(x) = \sum_{S \subseteq N} m_S \phi_S(x_S)
\]

where \( m_S \) are Möbius masses and \( \phi_S \) aggregates the quantities \( x_i, i \in S \) to define the interaction term \( \phi_S(x_S) \). Thus \( \phi_S \) is the product if \( F \) is the multilinear model and \( \phi_S \) is the min (resp. max) operation if \( F \) is the conjunctive (resp. disjunctive) form of the Choquet integral. Note that function \( F(x) \) reads as the following inner product \( F(x) = (m, \phi(x)) \) where \( m = (m_S)_{S \subseteq N} \) and \( \phi: \mathbb{R}^n \rightarrow \mathbb{R}^2 \) maps \( x \) into a nonlinear feature space: \( \phi(x) = (\phi_S(x_S))_{S \subseteq N} \). Both vectors \( m \) and \( \phi(x) \) are indexed by the subsets \( S \subseteq N \) numbered in lexicographic order.

Möbius masses and interactions. A capacity \( v \) is said to be \( k \)-additive if \( m_v(S) \neq 0 \) for some \( S \) of size \( k \) and \( m_v(S) = 0 \) for all \( S \) of size greater than \( k \) [Grabisch, 1997]. When \( v \) is 1-additive, all interaction terms vanish and \( F \) boils down to a weighted arithmetic mean of components \( x_i \). When \( v \) is 2-additive, only pairwise interactions are possible in the model (terms of type \( \phi_{ij}(x_i, x_j) \) in Equation 7). Their coefficients \( m_v(\{i,j\}) = v(\{i,j\}) - v(\{i\}) - v(\{j\}) \) can be positive or negative and their magnitude measures the importance of the interaction between \( i \) and \( j \) (the gap to additivity). It is frequently assumed that capacities are \( k \)-additive for some \( k < n \) prior to preference analysis so as to make sure that \( v \) admits a polynomial-size representation in terms of Möbius masses. However, it strongly reduces the descriptive power of the ML and Choquet models. For example, the preference for balanced solutions mentioned in the introduction requires \( m_v(N) > 0 \) (a bonus is given to alternatives that perform well on all criteria) which is incompatible with \( k \)-additivity for any \( k < n \). In this case, a \( n \)-additive capacity is needed without any restriction on the size of possible interactions.

Several measures of interaction have been proposed in the literature to measure the magnitude of interactions within sets of any size. For example, in the Banzhaf and Shapley interactions indices \( I_B(S) \) and \( I_{SB}(S) \) associated to a subset \( S \subseteq N \) in models \( M_L v \) and \( C_v \) are related to Möbius masses \( m_v \) as follows [Grabisch et al., 2009]:

\[
I_B(S) = \sum_{T \supseteq S} \frac{m_v(T)}{2^{|T|-|S|} + 1} \quad I_{SB}(S) = \sum_{T \supseteq S} \frac{m_v(T)}{|T|-|S| + 1}
\]

We remark that these interaction indices tend to vanish as Möbius masses tend to 0. Thus, favouring the sparsity of the Möbius vector \( m \) in an aggregation function defined by Equation 7 tends to reduce the criteria interactions in the associated decision model.

Related work. As far as the identification of the capacity used in a decision model is concerned, several approaches based on the least squares criterion or variance minimization of the model under preference constraints have been proposed in the field of multicriteria analysis for the Choquet integral [Grabisch et al., 2008; Grabisch and Labreuche, 2010]. In the field of machine learning different learning algorithms have been proposed, e.g., Choquistic regression [Tehrani and Hüllermeier, 2013], support vector machines (SVM) with Choquet kernel [Tehrani, 2021], and ridge regression for Choquet regression [Kakula et al., 2020]. Moreover a neural network was recently proposed to learn a hierarchical Choquet model [Bresson et al., 2020]. Active learning approaches also exists based on regret minimization [Benabou et al., 2017]. Some recent contributions using regression also exist for the multilinear model [Pelegrina et al., 2018; Pelegrina et al., 2020].

The complexity of models involving capacities, having in general \( 2^n - 2 \) free parameters, is prohibitive for many real-life applications. Very often, a prior complexity reduction is obtained by considering models with \( k \)-additive capacities (\( k = 2 \) being the most common choice) [Grabisch et al., 2008; Hüllermeier and Tehrani, 2013; Galand and Mayag, 2017; Ah-Pine et al., 2018; Bresson et al., 2020; Pelegrina et al., 2020]. Similar restrictions exist for limiting interactions with the notion of \( k \)-interactivity [Beliakov and Wu, 2019].

A less restrictive and more flexible attempt to reduce models complexity is to derive a sparse representation of the capacity from preference data using the \( L_1 \) penalty term [Anderson et al., 2014; Adeyeba et al., 2015; Pinar et al., 2017; de Oliveira et al., 2022] where the regularization was applied either to the capacity, or to the interaction index. Recently, some evidence was given that Möbius representations often
lead to more compact preference representations and an approach based on linear programming (LP) was proposed to learn a sparse Möbius transform of the capacity in the Choquet integral [Herin et al., 2022a]. However, the absence of prior restriction on capacities comes at the expense of computation times and scalability.

Our contribution in this paper is to propose a faster and more scalable algorithm to learn sparse Möbius representations of capacities from preference examples, for the multilinear and Choquet models and any other instance of Eq. 7. Our approach relies on iteratively re-weighted least squares and dualization as explained in the next section.

3 A Dual IRLS for Sparse Preference Learning

Our objective is to learn a sparse representation of $m$ based on a training set of preferences statements $\{(x^i, y^i) \in X^2 : x^i > y^i, i \in P\}$ and possibly of indifference statements $\{(x^i, y^i) \in X^2 : x^i \sim y^i, i \in I\}$. A well-known workhorse for learning sparse models is the $L_1$-norm penalty. This is indeed a sparse-inducing penalty, in the sense that it promotes for learning sparse models is the $L_1$-norm penalty. The constraint $\ell_1$-penalization is only applied to the terms involving at least two criteria so as to minimize interactions. The approximation problem is thus formulated as follows:

\[
\begin{align*}
\mathcal{P} & \quad \min \sum_{i \in P} \epsilon_i + \sum_{i \in I} (\epsilon_i^- + \epsilon_i^+) + \lambda \sum_{j=n+1}^{2^n} |m_j| \\
& \quad (m, \phi(x^i)) - (m, \phi(y^i)) + \epsilon_i \geq \delta, \quad i \in P \tag{8} \\
& \quad (m, \phi(x^i)) - (m, \phi(y^i)) + \epsilon_i^+ - \epsilon_i^- = 0, \quad i \in I \tag{9} \\
& \quad (m, 1) = 1 \quad \tag{10} \\
& \quad \epsilon_i \geq 0, \quad i \in P, \quad \epsilon_i^+, \epsilon_i^- \geq 0, \quad i \in I \tag{11}
\end{align*}
\]

where variable $m_j$ is the $j$th component of vector $m$. The hyper-parameter $\lambda > 0$ controls the level of regularization and $\delta$ is a strictly positive discrimination threshold used to separate preference from indifference situations. Variable $\epsilon_i$ models the positive error made on the preference example $x^i > y^i$, while $\epsilon_i^- - \epsilon_i^+$ models the signed error made on the indifference $x^i \sim y^i$. The constraint $m(0) = 0$ is implicit.

Problem $\mathcal{P}$ can be solved by linear programming using the following $L_1$-norm linearization:

\[
\begin{align*}
\mathcal{P} & \quad \min \sum_{i \in P} \epsilon_i + \sum_{i \in I} (\epsilon_i^- + \epsilon_i^+) + \lambda \sum_{j=n}^{2^n} (w_j^+ + w_j^-) \\
& \quad m_j = w_j^+ - w_j^-, \quad j = n + 1, \ldots, 2^n \\
& \quad w_j^+, w_j^- \geq 0, \quad j = n + 1, \ldots, 2^n \tag{12}
\end{align*}
\]

where variables $w_j^+, w_j^-$ are used for the linearization of $|m_j|$. Despite a simple linearization, the obtained linear program still drags an exponential number of variables $(2^n(3 - 2n) + |P| + 2|I|)$ and thus is hardly solvable for more than a dozen of criteria. For the sake of scalability, we propose to solve $\mathcal{P}$ by solving a sequence of sub-problems $\mathcal{P}_k$ that admit an efficient dual formulation. More precisely, we use an iteratively reweighted least square (IRLS) algorithm [Daubechies et al., 2010; Bach et al., 2012; Beck, 2015; Grandvalet, 1998] that consists in approximating the solution of a $L_1$-penalized problem with a sequence of least squares problems. Sparsity is recovered by increasingly penalizing non significant coefficients with a squared $L_2$ regularization.

\[
\begin{align*}
\sum_{j>n} |m_j| = \frac{1}{2} \min_{\eta \geq 0} \sum_{j>n} \frac{m_j^2}{\eta^2} + z_j \tag{12}
\end{align*}
\]

Using Eq. 12, we now establish Proposition 1 providing an IRLS algorithm that approximatively solves $\mathcal{P}$. The proof relies on the framework introduced in [Beck, 2015] that gives conditions under which an optimization problem can be solved by alternating minimization (here on $m$ and $z$) and insights on how this algorithm can lead to IRLS sequences.

Proposition 1. Let $\eta > 0$ be a smoothing parameter. Consider the sequence $m^{(k)}$ initialized with $m^{(0)} = 1$ such that:

\[
\begin{align*}
m^{(k+1)} & \in \operatorname{argmin} \sum_{i \in P} \epsilon_i + \sum_{i \in I} (\epsilon_i^+ + \epsilon_i^-) + \lambda \sum_{j>n} \frac{\lambda m_j^2}{|m_j|^2 + \eta^2} \\
& \quad \text{s.t.} \; (8), (9), (10), (11)
\end{align*}
\]

Then we have: $\lim_{k \to \infty} J(m^{(k+1)}) - J^* \leq (2^n - n) \eta$ where $J$ is the objective function of $\mathcal{P}$ and $J^*$ its optimum. $\mathcal{P}_k$ refers to the problem solved at each iteration.

Proof. Let $\eta > 0$ be a smoothing parameter and $\mathcal{P}_k$ the associated surrogate problem of $\mathcal{P}$ where the sum of absolute values is replaced by a differentiable term:

\[
\begin{align*}
\mathcal{P}_k & \quad \min \sum_{i \in P} \epsilon_i + \sum_{i \in I} (\epsilon_i^+ + \epsilon_i^-) + \lambda \sum_{j>n} \sqrt{m_j^2 + \eta^2} \\
& \quad \text{s.t.} \; (8), (9), (10), (11)
\end{align*}
\]

Remarkingly that $\epsilon_i = (\delta - \langle m, \delta^i \rangle)_+$ and $\epsilon_i^+ + \epsilon_i^- = (m, \delta^i)$ at the optimum, where $\delta^i = \phi(x^i) - \phi(y^i)$ and $(x)_+ = \max(0, x)$, $\mathcal{P}_k$ can be reformulated in an unconstrained form:

\[
\begin{align*}
\mathcal{P}_k & \quad \min \sum_{i \in P} \langle \delta - m, \delta_i \rangle_+ + \sum_{i \in I} |\langle m, \delta_i \rangle| \\
& \quad + \lambda \sum_{j>n} \sqrt{m_j^2 + \eta^2} + I_{\{m, \delta\}_{=1}} \\
\end{align*}
\]

with $I_{\{m, \delta\}_{=1}} = 0$ if $\langle m, \delta \rangle = 1$ and $+\infty$ otherwise. Then, introducing the smoothing parameter $\eta$ in Eq.12 yields:

\[
\lambda \sum_{j>n} \sqrt{m_j^2 + \eta^2} = \min_{\eta \geq 0} \frac{\lambda}{2} \sum_{j>n} \frac{m_j^2 + \eta^2}{z_j}
\]
which leads to reformulate $\mathcal{P}_\eta$ as a problem involving two blocks of variables $(m, z)$:

$$
(\mathcal{P}_\eta) \quad \min_{m, z} \quad H(m, z) = g_1(m) + g_2(z) + f(m, z)
$$

with

$$
g_1(m) = \frac{1}{2} \sum_{j>n} |m_j|^2 + z_j,
$$

$$
g_2(z) = \sum_{i \in P} (\delta^i - \langle m, \delta^i \rangle) + \sum_{i \in I} |(m, \delta^i)| + \lambda_1 \left( \sum_{i \in I} |(m, \delta^i)| \right)
$$

Since $g_1, g_2$ are closed proper convex functions sub-differentiable over their domains $dom\ g_1$ and $dom\ g_2$, and $f$ is convex and continuously differentiable over $dom\ g_1 \times dom\ g_2$ (and $\nabla f$ Lipschitz continuous), this problem fits in the class of problem solvable by alternating minimization [Beck, 2015].

A proof of a non asymptotic sublinear convergence rate of the alternating minimization method in this case is given in [Beck, 2015] and guarantees:

$$
\lim_{k \to \infty} H(m^{(k+1)}, z^{(k+1)}) - H^* = 0
$$

where $H^*$ is the minimal value of $H$. Since the optimization problem in Eq. 13 has the closed form solution $z^{(k+1)} = \sqrt{m^{(k+1)}_j^2 + \eta^2}$, this expression can be inserted in Eq. 14 which yields the following IRLS sequence:

$$
m^{(k+1)} = \arg\min_{m} g_1(m) + \sum_{j>n} \frac{\lambda m_j^2}{m_j^{(k+1)} + \eta^2}
$$

$$
eq \arg\min_{m} \sum_{i \in P} \epsilon_i + \sum_{i \in I} (\epsilon_i^+ + \epsilon_i^-) + \sum_{j>n} \frac{\lambda m_j^2}{m_j^{(k+1)} + \eta^2}
$$

s.t. (8), (9), (10), (11)

Let $J_\eta$ denote the objective function of $\mathcal{P}_\eta$. Using $|x| \leq \sqrt{x^2 + \eta^2} \leq |x| + \eta$ as in [Beck, 2015], we obtain:

$$
J(m^{(k+1)}) - J^* = J(m^{(k+1)}) - J_\eta(m^{(k+1)}) + J_\eta(m^{(k+1)}) - J^* + J_\eta(m^{(k+1)}) - J^* + J_\eta(m^{(k+1)}) - J^* = H(m^{(k+1)}, z^{(k+1)}) - H^* + (2n - n) \eta
$$

We conclude by passing to the limit and using Eq.15.

More precisely, since $\mathcal{P}_k$ is a convex problem with linear constraints, strong duality holds and there is no duality gap. Then solving $\mathcal{P}_k$ or its dual form is equivalent. The efficiency of the dual form of $\mathcal{P}_k$ is detailed in the following proposition:

**Proposition 2.** Problem $\mathcal{P}_k$ admits a dual formulation $\mathcal{D}_k$ which has $|P| + |I| + 1$ variables and $2(|P| + |I|)$ constraints:

$$
(\mathcal{D}_k) \quad \max_{\gamma=(\alpha, \delta, \mu) \in \mathbb{R}^{p+q+1}} \quad \Gamma^T \mathbf{D}_k \gamma - \frac{1}{4\lambda} \Gamma^T \mathbf{Q} \Gamma \mathbf{D}_k^{-1} \mathbf{Q} \Gamma + \Gamma^T \mathbf{L}
$$

where $p = |P|$, $q = |I|$ and $\mathbf{D}_k$ is a square diagonal matrix of size $2^n$ whose diagonal contains the current weighting coefficients $1/\sqrt{m^{(k)}_j^2 + \eta^2}$ (and $0$ for the singletons). Also, $\mathbf{Q}$ (respectively $\mathbf{L}$) is a data dependent matrix of size $2^n \times (p + q + 1)$ (respectively $p + q + 1$) such that $\mathbf{Q} = (\delta_\eta, \delta_q, \mathbf{1})$ where $\delta_\eta = (\delta^i_{\eta})_{i \in P}$ and $\delta_q = (\delta^i_{q})_{i \in I}$ are matrices of size $2^n \times p$ and $2^n \times q$ respectively and where $\delta_\eta = \delta_q = (1, 1, \ldots, 1) \in \mathbb{R}^p$ and $\mathbf{L} = (0, 0, \ldots, 0) \in \mathbb{R}^q$.

**Towards higher dimensions.** For a high number of criteria $n$, the computation of the matrix $\mathbf{Q}^T \mathbf{D}_k^{-1} \mathbf{Q}$ raises an issue since $\mathbf{Q}$ and $\mathbf{D}_k$ have $2^n$ columns. However, at the first iteration of the IRLS sequence, $\mathbf{D}_k$ is the identity matrix and the matrix $\mathbf{Q}^T \mathbf{D}_k^{-1} \mathbf{Q} = \mathbf{Q}^T \mathbf{Q}$ can be computed in polynomial time. In kernel-based machine learning, this property is referred to as the ‘kernel trick’ [Shawe-Taylor et al., 2004].

As computing the matrix $\mathbf{Q}^T \mathbf{Q}$ requires the computation of inner products of the form $\langle \phi(x), \phi(x') \rangle$, the ‘kernel trick’ reduces to a direct computation that does not require the calculation of vectors $\phi(x)$ (which are of size $2^n$ here). A computation in $O(n^2)$ is provided for the case of the Choquet integral $(\phi(x, y) = \min(x, y))$ in [Tehran et al., 2014]:

$$
(\phi(x), \phi(x')) = \langle x, x' \rangle + \sum_{i=1}^{n-1} x_i \left\{ \sum_{j=1}^{n-i} 2^{n-i-j} \cdot \min \left\{ x'_i, x'_{i+j} \right\} \right\}
$$

where $(.)$ is a permutation of $N$ such that $x_{i(1)} \leq x_{i(1)+1} \leq \ldots \leq x_{i(n)}$ are permutations sorting each vector $(x'_i, \ldots, x'_{i+n})$ by increasing order. This formula can also be used to obtain a polynomial computation of $(\phi(x), \phi(x'))$ when $\phi(x, y) = \max(x, y)$ since $\max(x) = -\min(-x)$. In addition, we provide a polynomial computation of the multilinear kernel $(\phi(x, y) = \prod_{i \in S} x_i)$ in the following proposition:

**Proposition 3.** Proposition 3 (See also [Shawe-Taylor et al., 2004]). When $\phi_S(x) = \prod_{i \in S} x_i$, we have: $(\phi(x), \phi(x')) = \sum_{S \subseteq N} \prod_{i \in S} x_i \prod_{i \not\in S} x'_i = \prod_{i=1}^n (x_i x'_i + 1) - 1$. Then $(\phi(x), \phi(x'))$ can be computed in $O(n)$.
Enforcing monotonicity. Even if monotonicity constraints on the capacity are omitted, it is likely that the learning algorithm captures the monotonicity of the preference examples. It has been observed in practice with the Choquet kernel SVM [Tehran, 2021] where the learned models achieve low monotonicity violation rates. However, if for normative reasons, we must guarantee that monotonicity w.r.t weak Pareto-dominance holds for all possible alternatives, hard monotonicity constraints must be put on the capacity. For $n$ criteria, there are $C(n) = \sum_{i=1}^{n} k_i$ monotonicity constraints that read in terms of Möbius masses, as follows: $m_{T \subseteq S, S \supseteq S} = 0, \forall i \in S \forall S \subseteq N$. Including in $P$ this set of constraints induces a dual problem $D_k$ with $|P| + |I| + 1 + C(n)$ variables. Thus the dualization benefit is lost and one may prefer a direct solving of $P$ with LP. Still, the exponential number of variables is an obstacle to scalability and it gets even worse when an exponential number of constraints is added. Hence we propose to handle the monotonicity constraints throughout a generation constraint algorithm that allows an optimal solution to be reached while incorporating only a small portion of the entire set of constraints [Jünger et al., 1993]. The algorithm is initialized with a solution of $P$ found without monotonicity constraints. Then we iteratively insert the constraints violated in the current solution. The next section presents numerical evidences of the benefits of the proposed method.

4 Numerical Tests

In this section we present the results of numerical tests performed on synthetic preference data. We test the ability of our algorithm (denoted D-IRLS for dual IRLS) to learn a multilinear model or a Choquet integral for a growing number of criteria. We compare it to an exact solving of $P$ with LP (denoted ES). Preference data are generated through randomly drawn sparse Möbius vectors $m$ (verifying monotonicity constraints) and utilities vectors $x, y$ are uniformly drawn within $[0, 1]^n$. The overall values $u(x)$ and $u(y)$ are computed and perturbated with a Gaussian noise ($\sigma = 0.03$) before being classified as preference or indifference training examples. We set the size of the training sets to $|P| + |I| = 500$ and of the test sets to $|P| = 1000$. The regularization parameter $\lambda$ is set to $\lambda = 1$. All tests are conducted on a 2.8 GHz Intel Core i7 processor with 16GB RAM and we used the mathematical programming Gurobi solver (version 9.1.2). For the D-IRLS method, the smoothing parameter is set to $\eta = 10^{-50}$ and the algorithm terminates when $\|m^{(k+1)} - m^{(k)}\|_2 \leq 10^{-3}$. Also, coefficients with absolute values smaller than $10^{-5}$ are discarded at each iteration.

Training time and generalizing performance. In the first experiment, we generate 10 training/test sets and evaluate the average training time of both algorithms as well as the generalizing performances of the learned models (average preference inversion on a test set). In order to evaluate the scalability of our method we vary the number of criteria from $n = 7$ to $n = 22$. Figure 1 shows the results for the learning of the multilinear model (1a,1b) and for the Choquet integral in its conjunctive form (1c,1d). We observe that for both decision models ES does not provide any solution after $n = 17$. However D-IRLS allows more than 4 millions of coefficients ($n = 22$) to be learned in less than 450 seconds. In contrast we observe that the generalizing performances of the learned decision models obtained with D-IRLS and ES are comparable. Since the number of training preference examples is constant, the test error globally increases with the number of criteria for both methods. Finally, we can notice that the test errors obtained for the learning of $ML_0$ are higher than the one obtained for the learning of $C_v$.

Enforcing monotonicity. In a second experiment, we assess the computational efficiency of the constraint generation algorithm used to guarantee monotonicity. We use the same experimental setting as above and let $n$ vary from 6 to 15. We compare the exact solving of $P$ under all monotonicity constraints (denoted ESC) with the exact solving of $P$ with constraint generation (denoted ESG). Both are solved using LP. In Table 1 we compare $C(n)$ the total number of monotonicity constraints in ESC, and $\tilde{C}(n)$ the average number of constraints generated in ESG. We observe that ESG (including all contraints) is slower than ES and limited to $n = 12$. ESG performs significantly better (up to 15 criteria) due to the progressive introduction of monotonicity constraints. We observe that only a small fraction of the entire set of monotonicity constraints are inserted before reaching an optimal and fully monotonic capacity.

Comparison with $k$-additive models. The advantage of using sparse models with possible large interactions instead of $k$-additive models is illustrated in Table 2 where we compare our method (D-IRLS) to an exact solving of $P$ with $k$-additivity constraints for $k = 2$ (2-add) and $k = 3$ (3-add), still under the same experimental setting. The generalizing performance is significantly improved while computation times remain admissible (and even better for a large $n$).

<table>
<thead>
<tr>
<th>n</th>
<th>$\tilde{C}(n)$</th>
<th>$C(n)$</th>
<th>Time ESG</th>
<th>Time ESC</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>3.2±6.4</td>
<td>192</td>
<td>0.6±0.2</td>
<td>0.6±0.1</td>
</tr>
<tr>
<td>9</td>
<td>2.4±7.2</td>
<td>2304</td>
<td>4.2±1.9</td>
<td>18.0±4.6</td>
</tr>
<tr>
<td>12</td>
<td>151.9±222.2</td>
<td>24576</td>
<td>61.0±30.4</td>
<td>1212.6±247.6</td>
</tr>
<tr>
<td>15</td>
<td>2777.6±4326.5</td>
<td>245760</td>
<td>3448.6±5613.1</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 1: $C(n), \tilde{C}(n)$ and training times for ESG and ESC.

<table>
<thead>
<tr>
<th>n</th>
<th>D-IRLS 2-add</th>
<th>3-add</th>
<th>D-IRLS 2-add</th>
<th>3-add</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>0.04</td>
<td>0.10</td>
<td>0.06</td>
<td>0.13</td>
</tr>
<tr>
<td>12</td>
<td>0.04</td>
<td>0.17</td>
<td>0.23</td>
<td>0.56</td>
</tr>
<tr>
<td>16</td>
<td>0.06</td>
<td>0.22</td>
<td>0.49</td>
<td>1.87</td>
</tr>
<tr>
<td></td>
<td>345.13</td>
<td>346.78</td>
<td>347.56</td>
<td>349.34</td>
</tr>
</tbody>
</table>

Table 2: Average test error and training time over 10 simulations.

Mixing different models of interactions. Preference learning produces an instance of model $F^*(x)$ defined by Eq. 7. Möbius coefficients $m_S$ are learned for any given monotonic function $\phi_S$ defining the nature of interaction terms.
Several interaction functions may coexist in the same model with a possible benefit in terms of sparsity. As an illustration, we randomly draw preference examples (noised) based on the Hurwicz criterion:  
\[ h(x) = \frac{1}{2}(\min_{i \in N} \{x_i\} + \max_{i \in N} \{x_i\}). \]

Although the min (resp. max) term admits a sparse representation in the conjunctive form (Eq. 5) of the Choquet integral (resp. disjunctive form, Eq. 6) this is not the case of \( h(x) \) that includes both terms. This suggests extending the model defined in Eq. 7 to include simultaneously several instances of \( \phi(S) \) (like min and max). If we write \( v = v^\wedge + v^\vee \) with \( (v^\wedge, v^\vee) \) two sub-normalized capacities, we have \[ C_v(x) = C_{v^\wedge}(x) + C_{v^\vee}(x). \]

Then using the conjunctive form for \( C_{v^\wedge}(x) \) and the disjunctive form for \( C_{v^\vee}(x) \) we obtain: \[ C_v(x) = \sum_{S \subseteq N} (m_{v^\wedge}(S) \min_{i \in S} \{x_i\} + m_{v^\vee}(S) \max_{i \in S} \{x_i\}). \]

Then we compute a sparse representation of \( C_v \) possibly including both conjunctive and disjunctive terms by solving a variant of problem \( P \) using the double penalization term \[ \lambda_\wedge \sum_{j > n} |m_{v^\wedge}(j)| + \lambda_\vee \sum_{j > n} |m_{v^\vee}(j)| \] under the normalization constraint \( \sum_{S \subseteq N} (m_{v^\wedge}(S) + m_{v^\vee}(S)) = 1 \). On Figure 2 we provide the regularization path obtained for increasing values of \( \lambda_\wedge = \lambda_\vee \). As expected, a model including only two factors (min and max) is progressively emerging.

5 Conclusion
We have addressed the problem of preference learning with interacting criteria by considering a large class of capacity-based decision models including the multilinear utility and the Choquet integral, known for their expressiveness. We proposed a unified approach to learn the models of this class based on the search of sparse M"obius representations of capacities, leading to simple models with sparse interaction patterns. This approach applies to instances possibly involving more than 20 criteria and allows the most significant interaction factors to be identified within millions of possibilities. This represents a significant improvement compared to previous approaches limited to a dozen of criteria. Moreover, the sparsity pattern is revealed from preference examples instead of resulting from a prior cardinality-based simplification of interactions, which greatly enhances the descriptive possibilities. The main directions to extend this work are 1) going further in scalability (a larger use of kernels is worth investigating) and 2) extending the approach to learn interaction functions \( \phi(S) \) from preference data (model selection problem).
Acknowledgments
This work is supported by the ANR project ANR-20-CE23-0018 THEMIS of the French National Research Agency.

References


